NUMERICAL SOLUTIONS TO TRANSONIC FLOW PAST AIRFOILS AND SLENDER BODIES BY THE INTEGRAL EQUATION METHOD
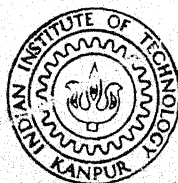
By

K. S. RAVICHANDRAN

ME TH
1983   ME/1983/D
D   R197n

RAV

NUM   **DEPARTMENT OF MECHANICAL ENGINEERING**

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JUNE, 1983

NUMERICAL SOLUTIONS TO TRANSONIC FLOW PAST AIRFOILS AND SLENDER BODIES BY THE INTEGRAL EQUATION METHOD

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

By

K. S. RAVICHANDRAN

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JUNE, 1983

ME-1983- D-RAV-NUM

CERTIFICATE

This is to certify that the thesis 'Numerical

Solutions to Transonic Flow Past Airfoils and Slender

Bodies by an Integral Equation Method' submitted by

K.S. Ravichandran in partial fulfilment of the degree

of Doctor of Philosophy to the Department of Mechanical

Engineering of I.I.T. Kanpur is a record of bonafide research

work carried out by him under our supervision and guidance.

The results embodied in this thesis have not been submitted

to any other University or Institute for the award of any
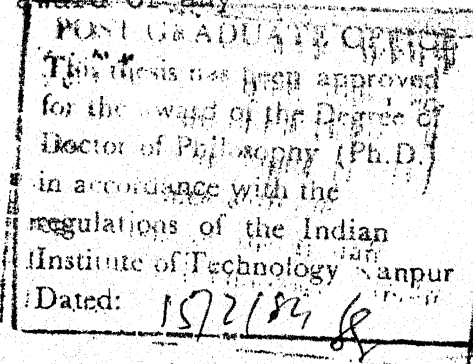
diploma or degree.

Thesis Supervisor:

N.L. Arora
Professor
Department of Aeronautical Engineering
Indian Institute of Technology
Kanpur

Co-Supervisor:

R. Singh
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology
Kanpur

ACKNOWLEDGEMENTS

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

List of Principal Symbols in Chapter II

x_1, r_1, θ_1 x, r, θ x', r', θ'	Cylindrical co-ordinates
c	velocity of sound
Φ	velocity potential
M_∞	free-stream Mach number
γ	ratio of specific heats
$\tau \ (= 1/FR)$	maximum body thickness
p	porosity factor
C_p	pressure co-efficient
$\bar{R}$	axisymmetric body shape function
ε	artificial viscosity co-efficient, perturbation parameter
s	cross-sectional area function for slender body
ϕ_1	perturbation potential on physical plane
ϕ	perturbation potential on reduce plane
W	test function
Ω	volume space for the application of Green's theorem
u_1, v_1	perturbation velocity components (physical plane)
β	$= (1 - M_\infty^2)^{1/2}$
K	$= (\gamma + 1) M_\infty^2$
n	normal
ψ	fundamental solution for Laplace equation
S	boundary of the region Ω

R	residual
S_B	body surface
S_p	infinitesimal sphere surrounding the point P
S_∞	the surface at infinity for free air flow
S_w	tunnel wall surface
$\bar{g}$	field source term
I	value of an integral
l	body length (≤ 1)
A, B, C, D	functions appearing in the wall integral
a, b, k	parameters in calculation of wall integral
E, F	elliptic function of first and second kind respectively
a_F, a_w	field and wall influence co-efficients respectively
δ	field-panel semi-width
h	field-panel height
∂	difference operator
J	an arbitrary grid function
f(u)	artificial viscosity function
α_1, α_2	co-efficients for the artificial time-dependent terms
ω	damping factor for the correction vector
C	correction matrix for quasi-Newton scheme

Abbreviations

FR	fineness ratio
SCH	scheme
SO	shock operator
JA	Jameson shock operator
EL	elliptic shock operator

Subscripts

i, j, k, l	indices
L	linear
F	field
w	wall

List of Principal Symbols in Chapter III

∇	gradient operator
ρ	density
Φ	total potential
c	velocity of sound
Q	fluid velocity
M_∞	free-stream Mach number
γ	ratio of specific heats
p	pressure
U, V	velocity components
g	field source term (inviscid)
α	angle of attack
λ	artificial viscosity parameter
$\emptyset$	perturbation potential
u, v	perturbation velocity components
$F_{u,l}(x)$	airfoil profile function
C_p	pressure co-efficient
x, y	co-ordinate axes
s, n	streamline co-ordinate system
μ	switching function
T	artificial viscosity term
G, H	artificial viscosity functions

$\bar{g}$	modified field source term
$\bar{\rho}$	modified density
$K_S,\ K_V$	kernel functions for a source and vortex respectively
$\sigma(x)$	source distribution
$\gamma(x)$	vortex distribution
$y_m(x)$	airfoil mean line
$\mathcal{H}$	symbol for functional notation
$\bar{x},\ \bar{y},\ \bar{z},\ w,$ $s,\ t,\ \nu,\ \theta$	Co-ordinate variables in transformations
r_t	leading edge radius
J	Jacobian
$e,\ f$	field influence co-efficients
$a,\ b,\ c,\ d$	influence co-efficients for internal singularities
C	matrix
$N,\ h$	vectors arising in discretized boundary condition
n	normal to body surface, No. of mean line elements
$\triangle\bar{g}$	change in $\bar{g}$
$\alpha_1,\ \alpha_2$	co-efficients for artificial time-dependent terms

Subscripts

$i,\ j,\ k,\ l$	indices
I	internal
F	field

SYNOPSIS

Among the numerical methods used for transonic flow computation, the Integral Equation Method (IEM) is one of the earliest. The non-linear integral equation for symmetric flow past a thin two-dimensional airfoil governed by the Transonic Small Perturbation Equation (TSPE) was first derived using Green's theorem in 1950. The IEM has since then been used for computing approximate numerical solutions to many transonic flow problems.

In this thesis entitled 'Numerical Solutions to Transonic Flows Past Airfoils and Alender Bodies by the Integral Equation Method', we are concerned with the application of the IEM for obtaining acceptable super-critical transonic flow solutions with shocks. The first problem considered is that of axisymmetric transonic flow past slender bodies of revolution with wind tunnel wall interference effects, satisfying the TSPE and linearized tangency boundary condition on the body surface and appropriate boundary conditions on the tunnel wall. Solutions were computed for general parabolic bodies of revolution by solving the non-linear integral equation numerically.

The success achieved in this attempt led to the idea that the IEM can be applied easily to obtain two-dimensional full-potential solutions for super-critical flows past arbitrary airfoils satisfying the exact boundary conditions. The present approach to this problem combines the idea of internal singularities for incompressible flows with the notion of a field source distribution representing the effects of compressibility

to formulate a system of two non-linear integral equations for the velocity components. Introducing artificial viscosity and upwind differencing, a simple direct iteration scheme is employed to obtain super-critical solutions with shocks for flows past arbitrary airfoils.

Chapter I of the thesis gives a review of the various numerical methods used at present for transonic flow calculations employing the potential flow equations. The Integral Equation Method, in particular, is reviewed in detail and the chapter concludes with a discussion of the motivation and scope of the thesis.

Chapter II presents the formulation of the problem for axisymmetric, transonic, wall-interference flow past slender bodies employing the small perturbation equation. The application of Green's theorem to the flow domain leads to the derivation of an integral equation for the axial perturbation velocity component. The integral equation is solved numerically by appropriate iterative procedures. The chapter concludes with a discussion of the computed solutions for flows past general parabolic bodies of revolution in free air as well as with porous will tunnel wall interference.

Chapter III presents the formulation of the problem for plane two-dimensional transonic flow past arbitrary airfoils, using the full-potential equation and exact tangency boundary condition. Two simultaneous non-linear integral equations are formulated for the two velocity components by considering the velocity induced at a field point by a distribution of singularities on the mean line of the airfoil together with a field

source distribution representing the compressibility effects. The integral equations are discretized on a curvilinear grid and solved numerically by a direct iteration scheme. Computed solutions for several test airfoils are presented in the concluding section of this chapter.

Chapter IV lists the conclusions to be drawn from the present work.

CHAPTER I

INTRODUCTION

1.1. GENERAL

The high lift to drag ratio of wings for free-stream Mach numbers just below the drag-rise Mach number [1] has long been recognized by aerodynamicists to make transonic flight design conditions an attractive proposition. However, it is precisely in this regime too that the flow has been recognized to be highly unstable. Early wind tunnel tests had already revealed that as the free-stream Mach number approached unity, the flow tended to become unstable leading to violent fluctuations in the surface pressure distribution and shock locations [2]. This posed problems for the stability of the aircraft crossing the transonic regime in flight. Compression shocks, which are typical features of transonic flow, may be strong enough to cause separation of the boundary layer leading to loss of lift and stalling of the aircraft. The high sensitivity of transonic flows to small changes in the flow parameters makes transonic wing design difficult. Detailed and accurate knowledge of the flow field throughout the transonic speed range is essential to specify reliable design and actual flight conditions. This can be acquired by either wind tunnel testing of models or computational schemes based on mathematical modelling of the real flow. In practice a judicious combination of both appears to be necessary.

Before the advent of the modern computers, aerody-
namicists relied mainly on wind tunnel data for studying the
flow fields around wing geometries for high subsonic and
transonic speed ranges. Wind tunnel testing has become very
expensive now. It was also recognized that wind tunnel simu-
lation of unbounded flow over the test body was seriously
impaired by wall interference effects at high subsonic free-
stream Mach numbers. Thus for high subsonic and transonic
flows, the development of analytical or numerical methods of
solution was particularly important. Analytical results for
transonic flows were obtained by using approximate methods to
solve the Transonic Small Perturbation Equation (TSPE) for
simple cases. However it is the development of faster and
larger computers in the sixties and seventies, which made
feasible for the first time, the computation of entire flow
fields by numerical methods. The cost of computation too
decreased remarkably, in inverse proportion to increasing
computer capabilities [3]. This gave the necessary impetus
to computational transonic aerodynamics in preference to costly
wind tunnel testing.

The computational methods for transonic flows may be
classified according to the choice of the governing equations
and the method of obtaining solutions. The governing equations
generally used are (i) the Navier-Stokes equations, (ii) the
Euler equations and (iii) the full-potential equation or the
TSP equation. The methods of solution may also be classified
under each of these headings. The three approaches differ

basically in the idealizing assumptions made about the real flow which then lead to the appropriate model equations.

In inviscid transonic aerodynamics, the chief feature of the flow is the appearance of weak shocks across which entropy rise is negligible. Crocco-Vaszonyi equation then states that such flows are also irrotational so that a velocity potential exists. Hence the equations describing inviscid compressible flow may be reduced to basically two equations, that of mass continuity and energy conservation. Although such a simplification is of enormous computational value, the results obtained by this approach, in the absence of appropriate entropy information, lead to uncertainties about the location and strength of calculated shocks. Calculations by Tai [4], using the Euler equations have shown that some of these uncertainties may be explained on the basis of Euler solutions assuming appropriate entropy increments across the shocks. In this thesis however, we shall confine ourselves to potential flow solutions obtained by numerical methods, keeping in mind their relevance to the Integral Equation Method (IEM).

A brief review of some of the methods that are widely used for computing transonic flows, is presented in this chapter. One of these, the integral equation method is employed in this thesis for computing axisymmetric and plane two-dimensional flows. A review of the IEM and a discussion of the scope of the thesis are presented later in the chapter.

1.2. FINITE-DIFFERENCE METHODS (FDM)

Most of the research efforts in transonic flow computation during the last decade has gone in to the development of finite-difference calculation procedures for two-dimensional and three-dimensional flow past wings, wing-body combinations etc. The pioneering work of Murman and Cole [5] in 1971, which virtually initiated the era of modern computational transonics, belongs to this category. They used a type-dependent, non-conservative differencing scheme to solve successfully the non-linear, type-mixed differential equation that describes small perturbation transonic flow past thin, two-dimensional, plane airfoils. Point and line relaxation techniques were used to obtain iterative solutions to the system of non-linear algebraic equations resulting from the differencing scheme. Shocks, if any, emerged naturally in the course of the iterative procedure, as regions of continuous flow with steep velocity gradients. Since then, relaxation schemes have been applied with great success to a wide range of transonic flow problems.

The mixed differencing scheme of Murman and Cole (1971) suffered from the drawback that it led to the use of an elliptic, centred difference operator at the first subsonic point downstream of the shock. This resulted in the shock jumps and shock locations being incorrectly calculated, the calculated strength being smaller than that predicted by the conservation laws. This deficiency was rectified by Murman in a later paper [6] in which the shock point operator was separately

introduced as the sum of the elliptic and hyperbolic operators
and this was shown to satisfy the conservation laws. It also
led to the correct jumps in the actual calculations.

The general method for constructing consistent diff-
erence schemes for hyperbolic systems of conservation laws
had been given earlier by Lax and Wendroff [7]. They had shown
that the correct jumps would be calculated only if the conser-
vation form of the equations was retained in the differencing
scheme. The mixed differencing scheme of Murman and Cole was
later put on a firmer footing by Jameson [8] and Albone [9],
who introduced the rotated difference scheme which makes the
differencing scheme virtually independent of the co-ordinate
system by correctly including the domain of dependence of the
differential equation in that of the difference equation (the
Courant-Friedrich-Lewy condition). Thus with the use of
conservative differencing together with type-dependent opera-
tors and rotated differencing at supersonic points, the problem
of computation of inviscid transonic flows may be said to have
been solved completely in principle. Later efforts in compu-
tational transonics have gone mostly into the development of
algorithms that are faster and more stable and also in deve-
loping satisfactory solution procedures for complicated three-
dimensional geometries.

The "speed of computation" problem has been approached
in several ways. In the initial stages of the development of
relaxation schemes, acceleration techniques akin to the
Aitken δ^2-process, developed by Hafez and Cheng [10] were
successful in achieving, at least in some cases, an order of

magnitude reduction in the number of iterations. The success of these techniques was however limited to a restricted class of flows. Other approaches incorporating hybrid iterative schemes have been tried by Jameson [11], giving encouraging results. In this method a fast Poisson solver is used after a specified number of relaxation sweeps to achieve faster convergence. The Poisson solver is unstable in the supersonic region and the relaxation sweeps are necessary to prevent instabilities from amplifying. The use of this technique is however limited by the strict conditions imposed on the grid spacing and the treatment of the boundary conditions upon which the efficiency of a Poisson solver critically depends. An alternative approach to the speed problem has employed the Approximate Factorization (AF) algorithms. The idea underlying this approach is that, if the linear operator used to evaluate the correction vector in an iterative scheme closely approximates the discretized partial differential operator used to evaluate the residual, the iterative scheme will converge rapidly. The AF algorithms seek to achieve this by expressing the linear operator as a product of factors each of which is easily invertible and which is a better approximation to the discretized partial differential operator than the unfactored one. A series of papers published in recent years indicate that this approach is superior to the line or point relaxation schemes [12-16].

A third approach engaging the attention of researchers has been the so called Multi Level Adaptive Technique (MLAT) [17-19]. In the MLAT, the discretization and solution processes

are intermixed. A sequence of uniform grids or levels with geometrically decreasing mesh sizes, participates in the process. The co-operative solution process on these grids involves relaxation sweeps over each of these grids, coarse grid to fine grid interpolation of corrections and fine grid to coarse grid transfer of residuals. Brandt and South [20] have applied the MLAT to the solution of TSPE and observed a high rate of convergence. Jameson [21] and McCarthy and Reynher [22] have recently used the method for transonic potential flow calculations, the former for flow past airfoils and wings and the latter for flow in axisymmetric inlets. Preliminary results indicate a substantial increase in the efficiency of the calculations. Some of the earlier difficulties experienced in the treatment of lifting flows and in calculations on non-uniform and curvilinear meshes have been tackled successfully by combining the multi-grid method with a generalized alternating direction method [21].

Falling within the category of finite-difference methods but differing from it by virtue of employing direct solvers, the semi-direct method developed by Martin and Lomax [23-24], appears to be a promising alternative for tackling the speed problem. In this approach the governing differential equation is formulated as a Poisson equation with the non-linear terms acting as forcing functions to which Murman's mixed difference operators are applied. The left side of the equation retains its elliptic character which, in the case of super-critical flows, is balanced by appropriate terms added to the right so that at convergence, the elliptic character inducing

terms cancel out on either side. Jameson [25], while giving
an excellent review of the finite-difference methods for tran-
sonic flows, has pointed out that such an elliptic operator on
the left is unstable for purely supersonic flows unless desymm-
etrized by an additional term. The addition of this desymm-
etrization term however destroys the regular structure required
by most direct solvers. This method is of particular interest
to us because the IEM casts the governing equation in the
Poisson form and transforms it into an integral equation. Thus
while sharing with the semi-direct method the disadvantages
arising out of the use of an elliptic operator, it differs
from it in the implementation of the body surface and far-field
boundary conditions which are now expressed as integrals over
appropriate bounding surfaces. This implies that the boundary
conditions in the IEM are satisfied relatively more simply and
accurately.

1.3. FINITE-ELEMENT METHODS (FEM)

Subsonic flow around airfoils and wings, formulated as
elliptic boundary value problems, have been calculated using
finite-elements by Argyris and Mareczek [26] and Periaux [27].
The presence of embedded supersonic zones and shock waves
complicates the application of finite-element methods to the
computation of transonic flows.

The basic idea common to most finite-element methods,
is that the problem of finding a solution to a boundary value
problem in partial differential equations is reformulated as
the problem of minimisation of a certain functional which may

or may not be subjected to constraints. In their review of the finite-element methods in transonic flows, Hafez et.al. [28] have classified these as hyperbolic, mixed and elliptic methods.

The hyperbolic method is applicable to hyperbolic partial differential equation systems. An approach using the Galerkin weighted residual technique was applied to the time-dependent transonic small perturbation equation by Wellford and Hafez [29]. Phares and Kneile [30] have applied a similar method to the equations of momentum conservation. Some preliminary results have been obtained using these methods but the convergence rate is rather disappointing compared to, for example, the finite-difference methods.

The mixed type methods are a finite-element analogue of backward differences, artificial time and artificial viscosity in the finite-difference methods. This method has been used by Hafez et.al. [31] who have calculated flows over a parabolic airfoil using rectangular elements and line relaxation technique. Chan and Brashears [32] have adopted a least squares approach to seek solutions to the TSPE by minimising a certain functional. This method fails to converge unless upstream effects are cancelled at supersonic nodes through a special assembly procedure.

Extensions of these methods to the full-potential equation are not trivial because here the flow direction is completely unknown. Difficulties therefore arise in the application of finite-element methods to super-critical flows involving change of strategy across the sonic line and shock wave.

Among the elliptic methods, Wellford and Hafez [33] adopt a variational approach and solve a minimisation problem for full-potential flow by standard finite-element techniques. They used a dual iteration method in which a direct Poisson solver is used for the potential function and a gradient method for the velocity function. An artificial viscosity was added to make the scheme converge. Results were obtained for a parabolic arc airfoil at super-critical Mach numbers.

Glowinski and Pironneau [34] have adopted an optimal control approach and solve a minimisation problem subject to constraints specified by the TSP or FP equation. They explicitly imposed the condition that the normal velocity downstream of the shock be less than that upstream and obtained results for symmetric flow past a NACA 0012 airfoil.

Eberle [35] has applied the finite-element method to the problem of airfoil optimisation in transonic flow. The addition of artificial viscosity enables the procedure to be converted into an analysis method. Solutions have been obtained by Eberle for flows past two-dimensional airfoils.

More recently Habashi and Hafez [36] have applied the finite element method to solve the two-dimensional, full-potential equation. Using Galerkin's weighted residual technique, and using artificial compressibility, they have obtained symmetric solutions for flows past NACA 0012 airfoil at subsonic and supersonic free-stream Mach numbers.

The Finite-Element method has also been used to obtain transonic flow solutions to the full-potential equation by reformulating the latter as an optimal control problem. Using

this approach, Bristeau [37] has obtained super-critical solutions for symmetric and lifting flows.

Although the finite-element method has not yet become competitive with finite-difference methods for transonic flow calculations, for problems involving geometrical complexity there is reason to believe that it will become so. For the TSP equation, using linear shape functions on triangle elements and line relaxation methods, Hafez et.al. [30] showed that finite-elements are comparable to Murman's finite-difference calculations. The validity of this conclusion in the case of the full-potential equation however remains to be seen. A more promising approach appears to be the finite-volume approach which is in fact a combination of the finite-element and finite-difference approaches and to which we will now turn our attention.

1.4. FINITE-VOLUME METHODS (FVM)

A major difficulty experienced in using finite-difference methods is that associated with the implementation of boundary conditions for different body geometries. For a Cartesian mesh, special finite-difference formulas are needed near the boundary leading to loss of accuracy. An alternative approach is to employ mappings to obtain geometrically simple boundaries on the computational domain. This may sometimes be a formidable task, especially for realistic three-dimensional geometries. A third approach, circumventing these difficulties, is provided by the finite-volume technique.

Finite-volume methods use general, non-orthogonal
co-ordinates and consider the governing differential equations
as balances of mass, momentum and energy fluxes for each
finite-volume defined by the intersection of co-ordinate
surfaces. This method was first introduced by MacCormack
and Paullay [38] who studied the discontinuous or weak solu-
tions of the time-dependent two-dimensional Euler equations
for supersonic flow past a wedge, a double symmetric wedge
and a sphere. Rizzi [39], Caughey and Jameson [40-42] have
all applied the finite-volume method to representative tran-
sonic flow problems. Caughey and Jameson use local bilinear
mappings to obtain curvilinear cells on the physical plane
from regular cubes on the computational plane. A similar
bilinear variation is assumed also for the velocity poten-
tial. They then use the usual finite-difference techniques
to cast the quasi-linear equations in difference form, while
ensuring that the flux balances across element interfaces
satisfy the mass conservation law.

The finite-volume method combines the advantages
of finite-elements for handling complicated geometries
with the simplicity of finite-differencing. Jameson and
Caughey [42] have reviewed the progress in the applica-
tion of finite-volume methods to calculation of tran-
sonic full-potential flows past general three-dimensional

wing-body combinations. They observe that though the solid surface boundary conditions are easily satisfied, satisfaction of far-field boundary condition suffers from the need to truncate the boundary at a finite distance from the body. This is a disadvantage that finite-volume methods still share with finite-difference techniques. Nevertheless, for many three-dimensional configurations of aerodynamic interest, grid construction is considerably simplified by the use of the finite-volume technique which requires co-ordinate transformations to be defined only locally. The accuracy in implementing the surface boundary condition is thereby improved without making the algorithm unreasonably complex.

1.5. INTEGRAL EQUATION METHODS (IEM)

In this thesis we are concerned with the use of IEM for the computation of transonic flows. A comparatively detailed review of this method is therefore given in this section.

The integral equation method is one of the earliest methods used to obtain solutions for transonic flows past airfoils, wings and bodies of revolution. In the IEM we deal with the flow directly on the physical plane. The potential flow equations are transformed into integral equations by the use of Green's theorem. This gives the perturbation potential at a general point in the flow field in terms of line or surface integrals over the flow boundaries (solid boundaries and flow discontinuities treated as internal boundaries) and a volume integral of all the non-linear terms lumped together and

acting as a field source distribution. The integral equation
is then discretized by dividing the flow field into panels or
cells and the resulting system of non-linear algebraic equations
is solved iteratively.

Most of the approaches to the transonic flow problem
by the IEM to-date have used the transonic small perturbation
equation, together with the linearized, mean-line/mean-surface
approximation for satisfying the tangency boundary condition.
Among the earliest such attempts, we must mention three related
studies by Oswatitsch, Gullstrand and, Spreiter and Alksne
(hereafter referred to as OGSA approach). Oswatitsch [43]
deduced a non-linear integral equation for a profile at zero
lift, resulting in a line integral around the airfoil and a
two-dimensional space integral. He treated shock waves as
internal boundaries, approximately normal to the airfoil surface
and across which the Rankine-Hugoniot jump conditions are satis-
fied. Introducing an assumed variation of the axial velocity
component in the transverse direction, he reduced the two-
dimensional problem to one with a single independent variable.
This problem was then solved by quadrature. This work was
later extended by Gullstrand [44 - 48] who also derived the
integral equation for symmetric flow past three-dimensional
wings and examined planar two-dimensional lifting flows.
Spreiter and Alksne [49] solved the integral equation for
sub-critical and continuous super-critical flows by iteration.
For flows with shocks, the shock location on the airfoil was
prescribed and the free-stream Mach number determined. In all

these cases, the method was restricted to two-dimensional symmetric flows past sharp nosed airfoils.

The method of Oswatitsch was later extended by Keune and Oswatitsch [50] to include the axisymmetric case. Heaslet and Spreiter [51] applied the IEM to transonic flow around slender wings and bodies of revolution. Special attention was given to the conditions resulting from the presence of shock waves and to the reduction of relations to special forms necessary for the discussion of flows at Mach one. Results were obtained for cone cylinders, wings and wing-body combinations at unit free stream Mach number and compared with experimental results. Klunker [52] deduced the integral equation for the velocity potential for transonic flow past thin lifting wings in subsonic free-streams, treating shock discontinuities appropriately. He also obtained the analytical expression for the far-field.

Nørstrud [53-56] extended the OGSA approach to lifting and non-lifting flows with shock discontinuities. The different functional approximations that can be made regarding the transverse variation of the axial velocity component were examined [53]. Applying the method of parametric differentiation, a Cauchy initial value problem was setup and solved by the numerical techniques of ordinary differential equations. This method however works only for shock free flows. For flows with shock discontinuities, the method of steepest descent was used. The initial velocity distribution in this case was assumed to have a discontinuity at an appropriate station. The solutions obtained were found to compare favourably with

finite-difference and experimental results. These techniques were extended to cover three-dimensional finite lifting wings [55-56]. Nixon and Hancock [57] derived two simultaneous integral equations for two-dimensional, planar lifting flows. Computed results showed close agreement with exact results [112] obtained by Sells. The method was applied by Nixon [58] to an airfoil oscillating at low frequency in a high Mach number subsonic flow.

Niyogi and his associates used the basic method of Oswatitsch to obtain approximate analytic solutions which are then used as input in an iterative solution scheme. These studies have been reported by Niyogi [59] in a review article. Niyogi [60] also sought to apply these techniques to three-dimensional finite symmetric wings at zero incidence and sub-critical speeds.

All the studies mentioned so far have dealt with the problem for free air flow only. Kraft and Lo [61] used the IEM for determining the effect of a perforated wind tunnel wall on the flow past a two-dimensional, non-lifting, plane airfoil at transonic speeds. They demonstrated that the correct shock location of free air flow can be obtained by the proper selection of porosity. Earlier Mokry [62] applied the IEM based on linear theory to compute subsonic flows past lifting two-dimensional airfoils in the presence of ventilated walls, assuming a general form of the classical wall boundary condition to simulate open jets, porous walls and slotted walls.

These early investigations using the IEM were severly restricted by computer capabilities available during the

sixties and early seventies. This limitation forced the investigators not to deal directly with the two-dimensional integral equation but to reduce it to one dimension, using arbitrary functional approximations for the transverse variation of the axial velocity component. This assumption is a feature common to all the methods discussed above. It is true that such a functional approximation, while drastically reducing the computational effort, does not seriously affect the accuracy of the final results at least for shock-free flows. For super-critical flows with shock however, the monotonic behaviour of the velocity implied by such functional approximations, breaks down in the vicinity of the shock, thus invalidating the whole procedure. Modifications to rectify these errors were suggested by Nixon [63] who introduced the 'extended integral equation method' wherein the flow field is divided into a finite number of streamwise strips and the transverse variation of the pertur-bation velocity across each of these strips is approximated by interpolation functions in terms of the values at the strip edges. Nixon [64-67] showed that this modification gave consi-derably improved results.

The first attempt to find exact sub-critical numerical solutions to the two-dimensional integral equation without any approximating assumptions, was made by Ogana and Spreiter [68-69]. In this approach, the flow field was divided into rectangular panels and the discretized system of non-linear algebraic equations was solved by Jacobi iteration [69]. Results were obtained for sub-critical flows past symmetric airfoils. Arora and Agarwal [70] used the field panel technique for the

problem of sub-critical axisymmetric flow past slender bodies

of revolution while Arora [71] treated transonic flow past

lifting airfoils including wind tunnel wall interference effects.

Lobo and Sachdev [72] used higher order functional

approximations for the accurate numerical evaluation of the

singular field integral. They solved the problem of disconti-

nuous super-critical flow past symmetric profiles at zero angle

of attack by the quadratic iteration scheme.

Pier$ and Slooff [73] obtained solutions for super-

critical transonic flow past a symmetric parabolic airfoil by

the explicit introduction of an artificial viscosity term into

the TSPE. The field source term was subjected to conservative

differencing as in Jameson's scheme. A field panel approach

with a point source approximation for the square field panels,

was used to discretize the integrals and the resulting system

of equations was solved iteratively by quasi-Newton scheme.

Calculated results for a parabolic airfoil show good agreement

with the results of finite-difference calculations. Similar

ideas have been pursued by Lobo and Sachdev [74] who used

artificial viscosity to obtain smeared shock solutions which

were then used as the initial input to a quadratic shock-fitting

scheme, yielding sharp shocks satisfying the Rankine-Hugoniot

jump relations. An approach similar to that of Piere and

Slooff was used by Ravichandran, Arora and Singh [75] for axi-

symmetric flow past slender bodies in the presence of perforated

wind tunnel walls. Artificial time-dependent terms modeled on

Jameson's scheme [8] were added to impart stability to the

iterative scheme. This effort will be described in detail in the following chapter.

All the methods described so far employed the small perturbation equation together with the linearized, mean-line approximate boundary condition as the governing relations for obtaining the integral equations. An important advantage of the small perturbation theory is that it permits the application of the tangency condition on the body axis for two-dimensional and axisymmetric flows and on the plane of the wing for three-dimensional flows. In a finite-difference calculation procedure, such an approximation facilitates the use of a uniform Cartesian grid so that the need for special formulas or interpolations to satisfy the boundary condition on the body surface when the latter does not lie on a co-ordinate line or surface, is obviated. But the mean surface approximation is obviously not valid near the nose region of an airfoil. Moreover, the fact that most of the flow problems, even in inviscid aerodynamics, are of the singular perturbation kind, implies that the TSPE solutions are not uniformly valid throughout the region of interest. Further, since the small perturbation equations are dependent upon the choice of the perturbation parameter, their applicability is confined to a restricted class of flows and body geometries. Small perturbation theory is often "tuned up" by scaling the variables in the problems by powers of the free-stream Mach number to obtain more accurate approximations to solutions of the full-potential equation [76]. For supercritical flows however, the location and strength of the shock may be strongly dependent upon the values of the similarity

exponents. These exponents are not known apriori for specific
cases and in a sense they may even be said to depend upon the
solutions to be obtained [77]. Hence attempts to tackle more
general flow problems by the small perturbation approach are
beset with many difficulties and restrictive assumptions.
Finally, recent bench mark data obtained by Hinson and Budges
[78] have led to the conclusion that "full-potential solutions
are uniformly in better agreement with experiment than small
disturbance results". The development of larger and faster
computers has made it possible to dispense with the small
disturbance theory altogether in favour of the full-potential
equation and exact boundary conditions. This approach has also
made it possible to develop computer codes capable of handling
a wide class of flows so long as they satisfy the weak shock
assumption for isentropic, irrotational flows.

Exact integral equation formulation for sub-critical
flows using the full-potential equation has been studied by Luu
and Goulmy [79] and by Stricker [80]. Combining the classical
surface singularity approach with the field panel method,
Stricker has obtained solutions for sub-critical flows past
lifting airfoils. Comparison with Sells' [112] results show good
agreement. These methods however do not cover supercritical
flows with shocks. In the present work, an exact integral
equation formulation for transonic flow using the full-potential
equation has been employed for the computation of super-critical
solutions with shocks for flows past two-dimensional airfoils.

1.6. MOTIVATION FOR THE PRESENT WORK AND SCOPE OF THE THESIS

In the light of the discussion above, it appears that
the integral equation method could be a promising alternative
for the computation of atleast some classes of transonic flow
problems, particularly those involving complex geometries. In
viewing the IEM as a possible alternative to the finite-difference,
finite-volume and finite-element methods, one has two prime
consideration in mind: (i) easy and accurate implementation of
the boundary conditions and (ii) the possibility of obtaining
reasonably accurate solutions on relatively coarse grids in
fewer number of iterations.

It has already been observed in the discussion of the
finite-difference and finite-volume methods, that the main problems
encountered therein concern the implementation of the boundary
conditions. In flow problems the boundary conditions are usually
of three types. One is the body surface boundary condition.
This is usually tackled by globally mapping the physical domain
on to a rectangular domain as in finite-difference methods or
by using locally defined mappings as in finite-volume methods.
Although this results in boundaries conforming to co-ordinate
lines or surfaces in the computational domain, making it possible
to use simple difference formulas for the derivatives, the
necessity of using one sided (backward or forward) differences
may lead to loss of accuracy in the calculation of normal deri-
vatives on the boundary points. To overcome this difficulty,
extremely close spacing of grid lines in the vicinity of body
boundaries becomes essential. The second problem is related to

the far-field boundary condition. This is usually satisfied by prescribing an analytic far-field approximation at the outer boundary of the grid on the physical plane, or the entire physical domain is mapped on to a finite computational domain in which case the vanishing far-field conditions are directly satisfied. In the finite-volume approach where only local mappings are employed, the far-field conditions are assumed to prevail at a finite distance away from the body, to which the computational domain is then arbitrarily restricted. A third type of boundary condition sometimes arises across internal fluid boundaries such as wakes or trailing vortex sheets but we will not be concerned with them here.

The IEM, in this respect, offers an attractive alternative. Here the boundary conditions are implicitly satisfied by the integral equations in the form of integrals over bounding surfaces. Since integration is an error smoothening process and therefore inherently more accurate than that of finding derivatives by taking differences, the body surface boundary condition can be accurately satisfied more or less independently of the grid used for the overall computation. Also since the far-field boundary condition is implicitly contained in the integral equations, the computational domain can be justifiably limited to a finite distance around the body. Though co-ordinate transformations may be employed, as done in the present work, they merely serve to set-up a convenient computational grid and are used only for the purpose of evaluating the derivatives of flow quantities. Thus in terms of accuracy, flexibility and

ease of implementation the method appears to be of promise atleast for two-dimensional flows.

Some of the problems associated with the use of the IEM may also be recognized at the outset. One of these is the need to compute and store a large number of influence co-efficients. The consequent severe restriction on the number of grid points, though not a problem for sub-critical calculations, could prove unsatisfactory for the computation of super-critical flows where sharp shocks may have to be captured. Secondly, instability in the numerical scheme for strongly super-critical flows may be expected by the use of a Poisson type of iteration scheme which suggests itself naturally for the present formulation. It should be noted that such an iteration scheme has failed to produce convergent results even for slightly super-critical flows in finite-difference calculations [23].

In the present work the IEM has been used to find TSP and FP solutions to axisymmetric flow past slender bodies of revolution and plane flow past two-dimensional airfoils respectively. In this approach, an artificial viscosity term which enables shock capture is added explicitly to the governing differential equation. Non-linear integral equations for the perturbation velocity components are then formulated. The integral equations are discretized on a rectangular or a curvilinear grid. The discretized system of equations is solved numerically using an appropriate iteration scheme.

In dealing with the IEM, the present thesis is concerned with the application of the method for the computation of practical flows. Theoretical questions regarding the convergence

and stability of the numerical schemes and the satisfaction of conservation laws across shocks will not be dealt with. Further, though the thesis gives a good idea of the computational effort involved, a comparative evaluation of the IEM with other methods, particularly finite-difference methods, has not been provided for obvious reasons. Such an evaluation is of course necessary in the long run but of questionable value at present because, to quote Golberg [81], "where comparisons are made 'first-cut' programs for integral equations are put up against 'state of the art' codes for partial differential equations. Even so the integral equation may win out". In this sense, the present work is to be regarded still as an exploratory first-cut attempt at using the IEM for transonic flow calculations.

In what follows, in Chapter II we present the formulation as well as the results of the problem for axisymmetric flow past slender bodies of revolution, using the transonic small perturbation equation and the linearized body boundary condition. Chapter III presents the full-potential formulation and the solutions obtained for plane flow past two-dimensional airfoils. Chapter IV, the last, lists the conclusions to be drawn from the present work.

CHAPTER II

AXISYMMETRIC TRANSONIC FLOW PAST SLENDER BODIES-SMALL PERTURBATION SOLUTIONS

2.1. INTRODUCTION

In this chapter, the axisymmetric transonic small perturbation equation, together with the linearized boundary condition on the body surface and appropriate outer boundary conditions for wind tunnel or free air flow, is converted into an integral equation for the axial perturbation velocity component by the application of Green's theorem. The integral equation is then discretized into a system of non-linear algebraic equations by dividing the flow field in the meridian plane into rectangular panels wherein the flow variables may be assumed constant. Using a quasi-Newton or a direct iteration scheme, the system of non-linear algebraic equations is solved to obtain surface pressure distributions for slender, parabolic bodies of revolution for free air and perforated wall cases. Section 2.5 of this chapter discusses the results of the computations and the performance of the iteration schemes for the axisymmetric case. The chapter concludes with a brief mention of some TSP results obtained for plane symmetric flow past parabolic arc airfoils.

Before presenting the formulation of the problem, a few remarks of an introductory nature will be made on the question of wall interference in transonic flow testing.

2.1.1. Wall Interference Effects

The object of wind tunnel testing of aerodynamic configurations is to acquire data that would predict the performance of test bodies (wings, fuselages, wing-body combinations etc.) under actual flight conditions in an infinite, interference free domain. It has however been recognized since long that, even for low speed conditions, the finite test section size of wind tunnels produces effects on the flow around the test object which result in the measured pressure distribution being different from what it would be in free flight. Along with these effects, another typical feature of transonic flow also comes into play — that the disturbances in the transverse direction decay much more slowly and hence extend much farther than for low subsonic flows, thereby demanding a larger tunnel section. Thus for super-critical flows with embedded supersonic pockets, the effect of tunnel walls becomes more pronounced and must be accounted for. Hence arises the need for precise quantitative and qualitative evaluation of wall interference in wind tunnel testing in general and transonic wind tunnel testing in particular.

There are basically two approaches to the problem of wall interference in transonic wind tunnel testing. One is the adaptive wall concept [82], the basic idea of which is as follows: In any wind tunnel test of a model configuration, it is possible to measure, at a convenient control surface away from the model and say near the walls, two perturbation flow quantities, such as flow deflection and static pressure or axial and transverse velocity components. Using one of the

measured quantities as the boundary condition on the chosen control surface, the exterior flow is solved for by satisfying the governing equations. This yields a prediction of the other flow property. The predicted value is compared with the measured value to assess their compatibility. The wall conditions are then changed iteratively until the computed and measured values agree, implying interference free flow in the tunnel. The second approach is the correctable interference concept [83] in which the distribution of a single experimentally measured flow quantity along a conveniently chosen control surface far from the model is used as the outer boundary condition in a computational procedure to obtain the flow field around a model. The comparison of these results with the measured values given an assessment of non-potential effects, while comparison with free air computations indicates the presence of wall interference. The correctable interference concept hence obviates the need to model the tunnel wall boundary condition. However there is no way of applying precise quantitative corrections to tunnel data for obtaining interference free results.

Calculations to assess wind tunnel wall interference have been carried out by Bailey [84] and Sedin [85] for axisymmetric flows and Murman [86] and Kacprzynski [87] for two-dimensional flows. Bailey, Sedin and Murman have used the TSPE formulation together with the classical homogeneous wall boundary condition. Kaeprzynski, in addition to the TSPE formulation, has also used the full-potential equation. More recently Karlsson and Sedin [88] have presented a method for calculating transonic wind tunnel wall interference in an axisymmetric

slotted test section. They discuss the problem of designing optimum slot shapes for obtaining interference free conditions. Kraft and Lo [61] have used the IEM for studying blockage effects in a perforated wall transonic wind tunnel. In their calculation however, an approximating assumption regarding the transverse decay behaviour of the axial velocity component is made, which though perhaps reliable for subsonic shock-free flows, becomes totally unreliable for super-critical flows with shocks. This points to the need for considering the entire flow field and the full form of the governing integral equation in order to obtain reliable results.

In modelling the presence of porous wind tunnel walls, the wall boundary condition assumes crucial importance. There has been some disagreement among researchers regarding the ability of the classical homogeneous wall boundary condition to simulate the actual properties of perforated and slotted walls. The advantage of the correctable interference concept is, as already mentioned, that it does away with the need for such models. Stahara and Spreiter [89], based on the correctable interference concept, have carried out extensive calculation for axisymmetric flow past slender bodies to conclude that "such a procedure can be useful as a practical wall correction method for moderately variable or fixed geometry transonic wind tunnels where alteration of the tunnel wall is limited or impossible". In the absence of wall data however, a wall boundary condition such as the one mentioned above has to be used. For the perforated wall atleast, Jacocks [90] found that the classical homogeneous model is a reasonable representation. The

difficulty here however is that the porosity factor is not only a function of the flow parameters and wall geometry, but also a function of plenum and test section pressures, involving non-linear effects. The assumption that the porosity factor is constant along the wall, which has been made in the present calculations, is hence non-representative but makes the computation simpler and serves illustrative purposes only. The replacement of the constant by a slowly varying function of distance along the wall is a simple matter but to deal with a porosity factor which is a function of wall pressures may involve slightly different methods for the accurate evaluation of the wall integral.

2.2. PROBLEM FORMULATION

2.2.1. The Full-Potential Equation

We consider axisymmetric transonic flow past a slender body of revolution in cylindrical co-ordinates (x_1, r_1, θ_1) with the origin at the body nose and the x-axis parallel to the free-stream.

The partial differential equation for the velocity potential Φ of a compressible, inviscid, non-heat conducting flow of a perfect gas with only weak shock discontinuities can be derived from the conservation laws as

$$(c^2 - \Phi_{x_1}^2) \, \Phi_{x_1 x_1} + (c^2 - \Phi_{r_1}^2) \, \Phi_{r_1 r_1} + \frac{c^2}{r_1} \, \Phi_{r_1}$$

$$- 2\Phi_{x_1} \Phi_{r_1} \Phi_{x_1 r_1} = 0 \qquad\qquad (2.2.1)$$

where c is the sonic velocity defined by the energy equation

$$c^2 = \frac{1}{M_\infty^2} - \frac{\gamma - 1}{2}(\phi_{x_1}^2 + \phi_{r_1}^2 - 1) \qquad (2.2.2)$$

In Eqs. (2.2.1) and (2.2.2), all velocities have been made dimensionless by the free-stream velocity. The length scales are all referred to the body length. M_∞ denotes the free-stream Mach number and γ is the ratio of the specific heats.

The boundary condition to be applied on the body surface is that the flow velocity vector must be aligned with the local tangent to the body surface. If the body surface is prescribed as

$$r_1 = \tau \bar{R}(x_1) \qquad (2.2.3)$$

where τ denotes the maximum thickness of the body, the tangency condition can be written as

$$\frac{\phi_{r_1}}{\phi_{x_1}} = \tau \frac{d\bar{R}}{dx_1} \quad \text{on} \quad r_1 = \tau \bar{R}(x_1) \qquad (2.2.4)$$

The outer boundary condition is to be prescribed at the far-field for free air flow and, on the tunnel wall and far upstream for wall interference flow.

The free air far-field boundary condition can be written as

$$\phi_{x_1} \to 1, \quad \phi_{r_1} \to 0 \quad \text{as} \quad (x_1^2 + r_1^2)^{1/2} \to \infty \qquad (2.2.5a)$$

For wall interference flow, the classical homogeneous wall boundary condition for perforated tunnel walls may be written as [91]

$$\Phi_{r_1} + p(x_1)(\Phi_{x_1} - 1) = 0 \quad \text{on} \quad r_1 = r_{1w} \qquad (2.2.5b)$$

where r_{1w} represents the wall radius and $p(x_1)$ is the porosity parameter which relates the pressure difference across the wall to the flow through the wall. $p(x_1)$ in general must be determined experimentally for a particular wall geometry and free-stream Mach number. In addition to Eq. (2.2.5b), it is assumed that the axial velocity component vanishes far upstream and downstream

$$\Phi_{x_1} \to 1, \quad \Phi_{r_1} \to 0 \quad \text{as} \quad x_1 \to \pm \infty$$

The local pressure co-efficient may be obtained from

$$C_p = \frac{2}{\gamma M_\infty^2} [\{1 - \frac{\gamma - 1}{2} M_\infty^2 (\Phi_{x_1}^2 + \Phi_{r_1}^2 - 1)\}^{\frac{\gamma}{\gamma - 1}} - 1] \qquad (2.2.6)$$

2.2.2. The Small Perturbation Equation

For flow past slender bodies at free-stream Mach numbers close to unity, the transonic small perturbation equation may be obtained from Eq. (2.2.1) by the method of matched asymptotic expansions, using an expansion parameter ε which is a measure of the magnitude of the perturbations [92].

Assuming that $(1 - M_\infty^2) \sim O(\varepsilon)$, it can be shown that [92] $\varepsilon \sim O(\tau^2)$ for axisymmetric transonic flow past a body of thickness τ. Denoting the perturbation potential by $\emptyset_1$, the TSPE in physical variables can be obtained as

$$[1 - M_\infty^2 - (\gamma + 1)M_\infty^2 \emptyset_{1x_1}] \emptyset_{1x_1 x_1} + \frac{1}{r_1} (r_1 \emptyset_{1r_1})_{r_1} = 0$$

$$(2.2.7)$$

The first order slender body boundary condition consistent with the order of approximation of Eq. (2.2.7) is

$$\lim_{r_1 \to 0} (r_1 \phi_{1r_1}) = \tau^2 \bar{R} \frac{d\bar{R}}{dx_1} = \frac{s'(x_1)}{2\pi} \qquad (2.2.8)$$

where $\dot{s}(x_1) = \pi \tau^2 \bar{R}^2(x_1)$ is the body cross-sectional area and prime denotes derivative with respect to x_1.

The outer boundary condition Eqs. (2.2.5) now become,

free air flow: ϕ_{1x_1}, $\phi_{1r_1} \to 0$ as $(x_1^2 + r_1^2)^{1/2} \to \infty$

$$(2.2.9a)$$

wall interference flow: $\phi_{1r_1} + p(x_1)\phi_{1x_1} = 0$ on $r_1 = r_{1w}$

$$(2.2.9b)$$

$$\phi_{1r_1}, \phi_{1x_1} \to 0 \text{ as } x_1 \to \pm\infty$$

Equation (2.2.7), by virtue of its essential non-linearity and mixed character, correctly describes inviscid isentropic flow throughout the transonic regime. Embedded supersonic pockets and weak shock discontinuities across which the entropy rise is negligible, are typical of super-critical transonic flows. These are contained in the admissible class of solutions to the non-linear equation satisfying the following principle of weak solutions

$$\int_\Omega [W_{x_1} \{(1 - M_\infty^2)\phi_{1x_1} - \frac{\gamma + 1}{2} M_\infty^2 \phi_{1x_1}^2 \} + (\frac{W}{r_1})_{r_1} (r_1 \phi_{1r_1})] \, d\Omega = 0$$

$$(2.2.10)$$

for arbitrary test functions $W(x_1, r_1)$. Ω denotes the flow region exterior to the body. Equation (2.2.10) can be used to

show that the following jump condition is satisfied across a discontinuity [93]

$$[(1 - M_\infty^2) <u_1> - \frac{1}{2} (\gamma + 1)M_\infty^2 <u_1^2>] <u_1> + <v_1>^2 = 0$$

where $u_1 = \emptyset_{1x_1}$, $v_1 = \emptyset_{1y_1}$ \hfill (2.2.11)

and $<>$ denotes a jump in the quantity across the discontinuity.

Finally the first order approximation to the pressure co-efficient may be obtained from Eq. (2.2.6) as

$$C_p = - 2\emptyset_{1x_1} - (\tau \frac{d\bar{R}}{dx_1})^2 \hspace{2cm} (2.2.12)$$

2.2.3. Reduced Variables

For convenience we now apply the following transformations to the physical variables in Eq. (2.2.7)

$$x = x_1 \hspace{1cm} r = \beta r_1 \hspace{1cm} \emptyset = \frac{K}{\beta^2} \emptyset_1 \hspace{1cm} (2.2.13)$$

Here $\beta^2 = 1 - M_\infty^2$ and $K = (\gamma + 1)M_\infty^2$. With these transformations Eqs. (2.2.7) and (2.2.9) become

$$\emptyset_{xx} + \emptyset_{rr} + \frac{1}{r} \emptyset_r = \emptyset_x \emptyset_{xx} \hspace{2cm} (2.2.14)$$

$$\lim_{r \to 0} (r\emptyset_r) = \frac{K}{\beta^2} \frac{s'(x)}{2\pi} \hspace{2cm} (2.2.15)$$

free air flow: $\emptyset_x, \emptyset_r \to 0$ as $(x^2 + r^2)^{1/2} \to \infty$ \hfill (2.2.16a)

wall interference flow: $\emptyset_r + \frac{p(x)}{\beta} \emptyset_x = 0$ at $r = \beta r_{1w} = r_w$

\hfill (2.2.16b)

$$\emptyset_x, \emptyset_r \to 0 \text{ as } x \to \pm \infty$$

2.2.4. Super-critical Flow and Artificial Viscosity

It must be noted that Eq. (2.2.14) admits solutions with expansion discontinuities which are physically inadmissible. For uniqueness, directional property corresponding to "entropy inequality" must then be restored in numerical schemes.* For super-critical flows the directional bias can be introduced by using upwind differencing to represent the derivatives along the stream-wise direction at the supersonic points. The assumption of small perturbation theory implies that the velocity vector deviates little from the x-axis in the entire field. The characteristics in the supersonic region are then locally symmetric about the x-axis. Hence the use of upwind differencing in the x-direction ensures the correct domain of dependence at the supersonic points. Considering the two-dimensional TSPE in conservation form we can write

$$f_x + g_y = 0 \quad \text{with} \quad f = \emptyset_x - \frac{1}{2}\emptyset_x^2 \; , \quad g = \emptyset_y$$

The error in evaluating f_x one mesh width Δx backward is $-\Delta x f_{xx}$. Thus upwind differencing introduces a discretization error

$$- \Delta x \frac{\partial}{\partial x}[(1 - \emptyset_x)\emptyset_{xx}] = - \Delta x \frac{\partial}{\partial x}[(1 - u)u_x]$$

with $u = \emptyset_x$, which can be regarded as an artificial viscosity in the supersonic region.

Turning this idea around, an artificial viscosity term of the form

---

* Inclusion of artificial viscosity alone does not appear to ensure uniqueness of computed solutions. Multiple solutions have been obtained for plane flows even with the inclusion of artificial viscosity, see [111].

$$\varepsilon \, \frac{\partial}{\partial x} \, [f(u)u_x]$$

where $\varepsilon > 0$ and is $O(\Delta x)$, and $f(u)$ $[\equiv (u - 1)] > 0$ and is of $O(1)$ can now be explicitly added to the TSPE, representing the relative retardation of flow in the x-direction at the super-sonic points. Equation (2.2.14), with the addition of artificial viscosity can then be written in the modified form as

$$\varnothing_{xx} + \varnothing_{rr} + \frac{1}{r} \varnothing_r = \bar{g}(x, r) \qquad (2.2.17)$$

with the non-linear source term

$$\bar{g}(x, r) = uu_x - \varepsilon \frac{\partial}{\partial x} [f(u)u_x] \qquad (2.2.18)$$

Solution of the modified Eq. (2.2.17) replace the shock discontinuities in the solutions to Eq. (2.2.14) by continuous narrow regions with steep velocity gradients.

The choice of a suitable artificial viscosity function $f(u)$ in Eq. (2.2.18), depends on essentially two considerations: (i) that the captured shock be steep given the constraints on mesh size and the transition through the sonic point be smooth and (ii) that the iteration scheme be stable and the convergence rate reasonable.

For small perturbation flows past symmetric airfoils, several artificial viscosity functions and their asymptotic shock layer characteristics have been discussed by Piers and Slooff [73]. Among the many artificial viscosity functions tried in the course of the present work, only $f(u) = u - 1$ was found suitable for axisymmetric flow computations.

Equations (2.2.17) and (2.2.18) together with the boundary conditions Eqs.(2.2.15) and (2.2.16) completely specify the problem for axisymmetric transonic flow past slender bodies. An application of Green's theorem to the flow domain yields the integral equation for the perturbation velocity potential.

2.3. THE INTEGRAL EQUATION

Figure 2.1 presents a schematic representation of the problem specified in the previous section together with the domain of application of Green's theorem.

Green's theorem states that if $\emptyset$ and Ψ are two arbitrary functions that are differentiable twice within the volume Ω bounded by a surface S then,

$$\int_{\Omega} (\ \nabla^2 \emptyset - \emptyset \ \nabla^2 \Psi \ )d\Omega \ = \int_{S} (\emptyset \ \frac{\partial \Psi}{\partial n} - \Psi \ \frac{\partial \emptyset}{\partial n})ds \qquad (2.3.1)$$

where n denotes the normal drawn into Ω and ∇^2 denotes the Laplacian operator which, for the present case, in terms of the cylindrical co-ordinates is

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}$$

For the present problem $\emptyset$ is chosen as the function satisfying Eqs. (2.2.17), (2.2.15) and (2.2.16) while Ψ is assumed to be the fundamental solution of the Laplace equation given by

$$\Psi \ = \ 1/[(x - x')^2 + r'^2 + r^2 - 2rr' \cos(\theta - \theta')]^{1/2} \qquad (2.3.2)$$

Fig. 2.1. Schematic Representation of the Axisymmetric Problem and the Domain for the Application of Green's Theorem.

In Eqs. (2.3.2), (x, r, θ) denote the field point and (x', r', θ') denote the running variables. Note that Ψ is singular when (x', r', θ') coincides with (x, r, θ).

With the above choice of ϕ and Ψ, Eq. (2.3.1) holds in the region Ω of Figure 2.1 bounded by the surface S which includes the body surface S_B, an infinitesimal sphere S_p surrounding the field point P(x, r, θ) and the surface S_∞ for free air flow or the wall surface S_W for flow confined by tunnel walls. Since shock discontinuities in the flow are assumed to have been replaced by continuous regions with steep velocity gradients, the region as defined above satisfies the conditions of Green's theorem. Further the flow being axisymmetric, without loss of generality, we may take θ = 0 in Eq. (2.3.2). The various integrals occurring in Eq. (2.3.1) may be evaluated as follows.

The Volume Integral:

Since $\nabla^2 \Psi = 0$ and $\nabla^2 \phi = \bar{g}$ in the region under consideration we have from Eq. (2.3.1)

$$I_\Omega = \int_\Omega (\Psi \nabla^2 \phi - \phi \nabla^2 \Psi)\, d\Omega$$

$$= \iiint_\Omega \Psi\, (x,\, r;\, x',\, r',\, \theta')\, \bar{g}(x',\, r')\, r'\, d\theta'\, dr'\, dx'$$

$$(2.3.3)$$

The integral in Eq. (2.3.3) being convergent, the integral over the spherical cavity surrounding the point P makes negligible contribution in the limit. The region Ω may therefore be taken to be the entire space.

The Surface Integrals:

(i) The Integral Over S_p

The integral over the point cavity S_p may be evaluated
as follows. The point P is surrounded by a sphere of radius ε
centred at P. Letting $\varepsilon \to 0$

$$
\begin{aligned}
I_{S_p} &= \int_{S_p} (\emptyset \frac{\partial \Psi}{\partial n} - \Psi \frac{\partial \emptyset}{\partial n}) \, ds \\
&\quad \varepsilon \to 0 \\
&= \underset{\varepsilon \to 0}{Lt} (\emptyset(x, r) \frac{\partial \Psi}{\partial \varepsilon} - \Psi \frac{\partial \emptyset}{\partial \varepsilon}) \, 4\pi\varepsilon^2 \\
&= \underset{\varepsilon \to 0}{Lt} [- \emptyset(x, r) \frac{1}{4\pi\varepsilon^2} - \frac{1}{4\pi\varepsilon} \frac{\partial \emptyset}{\partial \varepsilon}] \, 4\pi\varepsilon^2 \\
&= - \emptyset(x, r) \qquad\qquad\qquad\qquad (2.3.4)
\end{aligned}
$$

(ii) The Integral Over S_B

The integral over the body represented by the slit S_B
can be evaluated as follows

$$
I_{S_B} = \int_{S_B} (\emptyset \frac{\partial \Psi}{\partial n} - \Psi \frac{\partial \emptyset}{\partial n}) \, ds
$$

When the body is sufficiently slender, the following
approximation may be made

$$
\underset{r' \to 0}{\lim} \frac{\partial}{\partial n} = \frac{\partial}{\partial r'}
$$

Therefore

$$
I_{S_B} = \int_o^1 \int_o^{2\pi} (\emptyset \frac{\partial \Psi}{\partial r'} - \Psi \frac{\partial \emptyset}{\partial r'}) \, r' \, d\theta' \, dx'
$$

where 1 is the location of the body sting. For bodies without
sting $1 = 1$. In the limit $r' \to 0$, $\emptyset$ is of the form

$$\emptyset = \frac{1}{2\pi} s'(x') \ln r' + g(x')$$

where $g(x')$ is a constant of integration (see [92]).

Also from Eq. (2.3.2)

$$\lim_{r' \to 0} \Psi_{r'} = \frac{r \cos(\theta - \theta')}{4\pi [(x - x')^2 + r^2]^{3/2}}$$

Hence in the limit as $r' \to 0$, $r' \emptyset \Psi_{r'} \to 0$ and the integral becomes

$$I_{S_B} = - \int_0^1 \int_0^{2\pi} \Psi \frac{\partial \emptyset}{\partial r'} r' \, d\theta' \, dx' \Big|_{r' \to 0}$$

The tangency boundary condition (2.2.15) may now be used to obtain

$$I_{S_B} = - \int_0^1 \int_0^{2\pi} \frac{K}{2\pi \beta^2} s'(x') \Psi \, d\theta' \, dx'$$

$$= - \frac{K}{4\pi \beta^2} \int_0^1 \frac{s'(x')}{[(x - x')^2 + r'^2]^{1/2}} \, dx' \qquad (2.3.5)$$

The integral in Eq. (2.3.5) may be recognized as giving the potential for linear subsonic flow past a slender body.

(iii) The Integral Over the Outer Boundary

(a) For the free air case, the outer boundary condition is to be applied at infinity. Hence the integral over the surface at infinity is given by

$$I_{S_\infty} = \int_{S_\infty} (\emptyset \frac{\partial \Psi}{\partial n} - \Psi \frac{\partial \emptyset}{\partial n}) \, ds$$

$$= - \int_{-\infty}^{\infty} \int_0^{2\pi} (\emptyset \Psi_{r'} - \Psi \emptyset_{r'}) r' \, d\theta' \, dx' \Big|_{r' \to \infty}$$

$$= 0 \qquad (2.3.6)$$

under the assumption that $\emptyset \sim \frac{1}{R^\epsilon}$, $\epsilon > 0$ and $R \to \infty$.

(b)   For the wall interference case, the integral over the cylindrical wall surface has to be evaluated.

$$I_{S_W} = - \iint_{S_W} (\emptyset \, \Psi_{r'} - \Psi \, \emptyset_{r'}) r_W \, d\theta' \, dx' \quad \text{at} \quad r' = r_W$$

Applying the wall boundary condition (2.2.16b) we get

$$I_{S_W} = - \iint_{S_W} (\emptyset \, \Psi_{r'} + \frac{p}{\beta} \, \Psi \emptyset_{x'})_{r'=r_W} r_W \, d\theta' \, dx'$$

After a few simple manipulations (see Appendix A(i)), we obtain

$$I_{S_W} = \int_{-\infty}^{\infty} [(A + B) \, \emptyset(x', r_W) - C \frac{p}{\beta} \emptyset_{x'}(x', r_W)] \, dx'$$

$$(2.3.7)$$

where

$$A = (r_W/\pi)(r_W - r) \, E(\pi/2, k)/[(a - b)a^{1/2}] \qquad (2.3.8a)$$

$$B = [F(\pi/2, k) - E(\pi/2, k)]/(2\pi a^{1/2}) \qquad (2.3.8b)$$

$$C = r_W \, F(\pi/2, k)/(\pi a^{1/2}) \qquad (2.3.8c)$$

$$a = (x - x')^2 + (r + r_W)^2, \quad b = 4 r r_W, \quad k = (b/a)^{1/2}$$

$$(2.3.8d)$$

and F and E are complete elliptic integrals of the first and second kind respectively.

The surface integral over the outer boundary can therefore be expressed as

$$I_{S_T} = \delta \int_{-\infty}^{\infty} [(A + B) \, \emptyset(x', r_W) - C \frac{p}{\beta} \emptyset_{x'}(x', r_W)] \, dx' \qquad (2.3.9)$$

where

δ = 0   for free air case

= 1   for wall interference case.

Using Eqs. (2.3.4) to (2.3.9) in Green's identity (2.3.1), we get the following integral equation for the perturbation velocity potential

$$\emptyset(x, r) = -\frac{K}{4\pi\beta^2} \int_0^1 \frac{s'(x') \, dx'}{[(x - x')^2 + r^2]^{1/2}}$$

$$+ \delta \int_{-\infty}^{\infty} [(A + B) \, \emptyset(x', r_w) - C \frac{p}{\beta} \, \emptyset_{x'}(x', r_w)] \, dx'$$

$$- \iiint\limits_{\Omega} \Psi \, \bar{g}(x', r') \, r' \, d\theta' \, dr' \, dx' \qquad (2.3.10)$$

The integral equation for the axial velocity component is obtained by differentiating Eq. (2.3.10) with respect to x (see Appendix A(ii)).

$$u(x, r) = \frac{K}{4\pi\beta^2} \int_0^1 \frac{x - x'}{[(x - x')^2 + r^2]^{3/2}} s'(x') \, dx'$$

$$+ \delta \int_{-\infty}^{\infty} (A + B + D) \, u(x', r_w) \, dx'$$

$$+ \iiint\limits_{\Omega} \Psi_{x'}(x, r; x', r', \theta') \, \bar{g}(x', r')$$

$$r' \, d\theta' \, dr' \, dx'$$

$$= u_L + \delta u_w + u_F \qquad (2.3.11)$$

where

$$D = \frac{p(x')}{\beta} \frac{x - x'}{r_w - r} A$$

and subscripts L, w and F stand respectively for the linear, wall and field contributions to the perturbation velocity.

Equation (2.3.11) is the non-linear integral equation for the axial component of the perturbation velocity whose solution must be obtained numerically. The numerical scheme for its solution is presented in the next section.

2.4. NUMERICAL SOLUTION

2.4.1. Discretization of the Integral Equation

In order to discretize Eq. (2.3.11), x-r plane is divided into rectangular panels (Figure 2.2a). Making a constant source approximation for each panel element, the field integral may be approximated by the sum

$$u_F(x, r) = \sum_{k} \sum_{l} a_F^{kl}(x, r) \, \bar{g}_{kl} \qquad (2.4.1)$$

where the indices k and 1 run along the x and r directions respectively.

Similarly the wall integral in Eq. (2.3.11) may be replaced by the sum

$$u_w(x, r) = \sum_{k} a_w^{k}(x, r) \, u_{kl}, \quad l = l_{max} \qquad (2.4.2)$$

The aerodynamic co-efficients a_F and a_w are defined as

$$a_F^{kl}(x, r) = + \int_{x_k-\delta}^{x_k+\delta} dx' \int_{r_l-h}^{r_l+h} r' \, dr' \int_{o}^{2\pi} \frac{\partial \Psi}{\partial x'} \, d\theta' \qquad (2.4.3)$$

$$a_w^{k}(x, r) = \int_{x_k-\delta}^{x_k+\delta} (A + B + D) \, dx' \qquad (2.4.4)$$
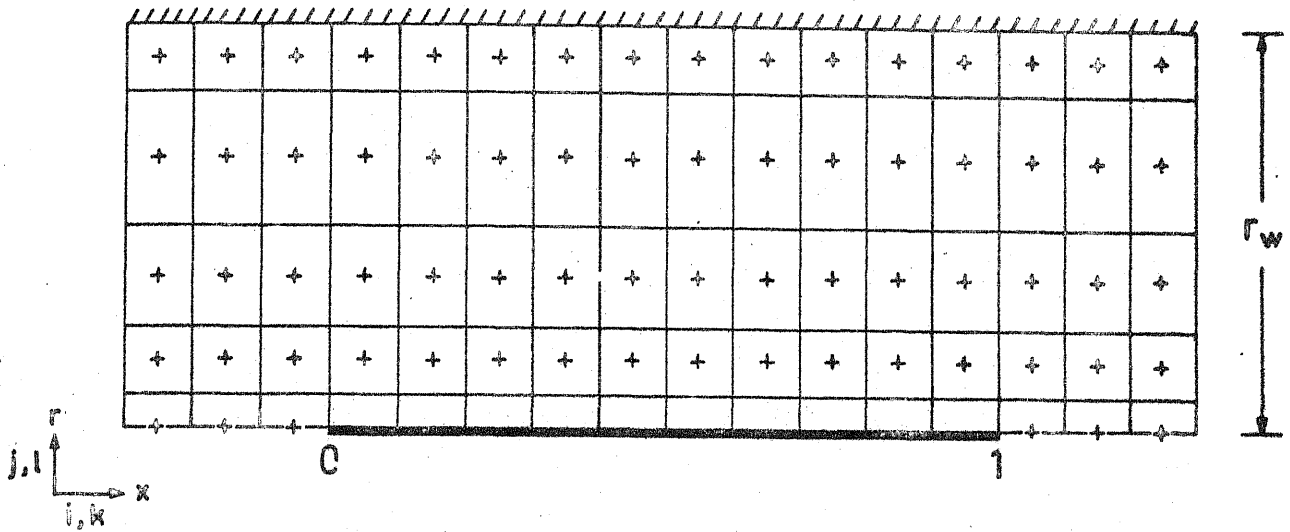
Fig. 2.2a.   The $x-r$ Plane Divided into Panels.
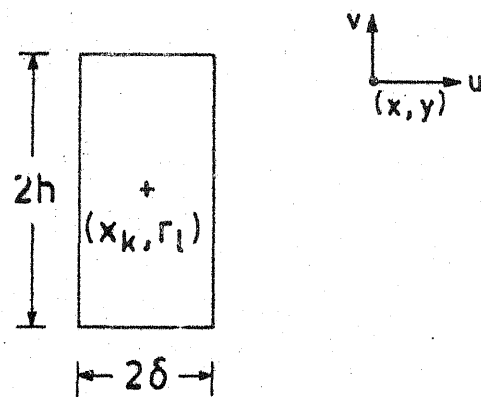
Fig. 2.2b.   Geometry for the Influence Coefficient
Calculation.
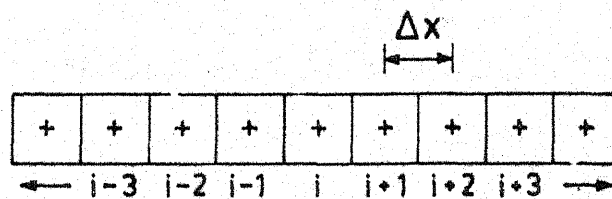
Fig. 2.2c.   Geometry for the Definition of the
Difference Operators.

where 2δ $(=\Delta x)$ and $2h$ respectively are the width and height of each panel element, see Figure 2.2b. The indicated integrations in Eqs. (2.4.3) and (2.4.4) are performed after approximating the values of the elliptic integrals occurring in the integrands by their values at the centre of the element. The expressions for a_F and a_W are given in Appendix B.

The first term in Eq. (2.3.11) represents the linear velocity distribution for subsonic axisymmetric flow past slender bodies. The integral occurring in this term must in general be numerically evaluated. A careful procedure is necessary for evaluating this integral for points close to the surface where the integrand has a singular behaviour. We have,

$$u_L(x, r) = \frac{K}{4\pi\beta^2} \left[\frac{s'(1)}{[(x-1)^2 + r^2]^{1/2}} - \int_0^1 \frac{s''(x')}{[(x-x')^2 + r^2]^{1/2}} dx' \right]$$

$$= \frac{K}{4\pi\beta^2} \left[\frac{s'(1)}{[(x-1)^2 + r^2]^{1/2}} \right.$$

$$+ s''(x) \ln \frac{x - 1 + [(x-1)^2 + r^2]^{1/2}}{x + (x^2 + r^2)^{1/2}}$$

$$\left. + \int_0^1 \frac{s''(x) - s''(x')}{[(x-x')^2 + r^2]^{1/2}} dx' \right] \qquad (2.4.5)$$

where $s'(o)$ has been assumed to be zero. The integral occurring in the third term of Eq. (2.4.5) is well behaved for points close to or on the surface and can be evaluated by quadrature.

2.4.2. The Difference Operators

In the discretization scheme to be adopted, we shall be using first and second order accurate difference operators

to evaluate the derivatives. They are defined as follows.

Let the grid lines parallel to the y-axis be spaced a uniform distance Δx apart in a Cartesian mesh and a function J be defined at the grid points, see Figure 2.2c. A second order accurate central difference operator on such a mesh may be defined as

$$[\partial_c^{(2)} J]_i = (J_{i+1} - J_{i-1})/2 \Delta x \qquad (2.4.6)$$

where the subscript c stands for central differencing.

Similarly other first order and second order accurate backward and forward difference operators may be defined as

$$[\partial_b^{(2)} J]_i = (3J_i - 4J_{i-1} + J_i)/2 \Delta x \qquad (2.4.7)$$

$$[\partial_f^{(2)} J]_i = (- 3J_i + 4J_{i+1} - J_{i+2})/2 \Delta x \qquad (2.4.8)$$

$$[\partial_b^{(1)} J]_i = (J_i - J_{i-1})/\Delta x \qquad (2.4.9)$$

$$[\partial_f^{(1)} J]_i = - (J_i - J_{i+1})/\Delta x \qquad (2.4.10)$$

where the subscripts b and f denote backward and forward differencing respectively.

2.4.3. The Non-linear Source Term

The function $\bar{g}(x, r)$ appearing in Eq. (2.2.17) represents the field source distribution. Different computational schemes may be formulated depending upon the form in which this term is written. In the present work three different representations for the evaluation $\bar{g}$ were tried. Denoting the schemes as S1, S2 and S3 we write,

$$\bar{g}(x, r) = \frac{\partial}{\partial x} (\frac{u^2}{2}) - \varepsilon \frac{\partial}{\partial x} [f(u)u_x] \quad : \quad S1 \qquad (2.4.11a)$$

$$= uu_x - \varepsilon \frac{\partial}{\partial x} [f(u)u_x] \qquad : \quad S2 \qquad (2.4.11b)$$

$$= uu_x - \varepsilon f(u)u_{xx} \qquad : \quad S3 \qquad (2.4.11c)$$

In [73], Eq. (2.4.11b) has been called a quasi-conservative scheme and Eq. (2.4.11a), a fully conservative scheme. Schemes S1, S2 and S3 as formulated above, are comple-tely equivalent for purely subsonic flows because the artificial viscosity term vanishes identically, while differencing of the inviscid term in Eqs. (2.4.11) in conservative and non-conser-vative form are both equivalent in the sense that they converge to the same solution in the limit as the mesh width goes to zero.

For super-critical flows with shock, the three schemes are capable of giving solutions which differ considerably in the shock region. This is so because the captured shock jumps are functions of the form in which the governing equations are differenced, the correct jump being the one obtained when the governing equations are differenced in a form expressing a conservation law. Since the inviscid equation in the present formulation is already written in the quasi-linear form, schemes S1 and S2, in which the artificial viscosity term is in conser-vative form, may be said to correspond to the quasi-conservative formulation while scheme S3, with the non-conservative viscosity term, may be said to correspond to the non-conservative quasi-linear formulation.

Introducing a switching function μ in the artificial viscosity term

$$\mu = 0 \qquad u < 1$$

$$\qquad (2.4.12)$$

$$= 1 \qquad u \geq 1$$

where $u = 1$ corresponds to sonic conditions, the field source term may be computed from

$$\bar{g}_{ij} = [\partial_c^{(2)} (\tfrac{u^2}{2})]_{ij} - \lambda [\partial_b^{(1)} (\mu \, f(u)u_x)]_{ij} \quad : \quad \text{S1} \qquad (2.4.13a)$$

$$= [uu_x]_{ij} - \lambda [\partial_b^{(1)} (\mu \, f(u)u_x]_{ij} \qquad\qquad : \quad \text{S2} \qquad (2.4.13b)$$

$$= [uu_x]_{ij} - \lambda \, f_{ij} [\partial_b^{(1)} (\mu \, u_x)]_{ij} \qquad\qquad : \quad \text{S3} \qquad (2.4.13c)$$

where $\lambda = \varepsilon / \triangle x$ and

$$u_{xij} = [\partial_c^{(2)} u]_{ij} \qquad \text{for interior points}$$

$$\qquad (2.4.14)$$

$$= [\partial_b^{(2)} u]_{ij} \quad \text{or} \quad [\partial_f^{(2)} u]_{ij} \quad \text{for boundary points}$$

The numerical viscosity parameter λ helps in regulating the magnitude of the dissipation term and hence the stability of the iteration scheme. Use of Jameson's switching function μ in conservative form in the artificial viscosity term leads to the correct sonic and shock point operators used first by Murman [6] in his conservative scheme.

In the original non-conservative scheme of Murman and Cole [5], use of type dependent differencing for the x-derivatives led to the use of an elliptic operator at the first subsonic mesh point downstream of the shock (to be referred to

hereafter as the shock point). This was later called the non-conservative scheme by Murman [6]. Since non-conservative results are often closer to experimental results, they are of some interest, although the shock jumps calculated by such schemes do not satisfy the conservation laws. Hence in the present work results have also been obtained by using the elliptic operator at the shock point. The field source term, at the shock point is accordingly calculated from

$$\bar{g}_{ij} = [\partial_c^{(2)} (\frac{u^2}{2})]_{ij} \quad : \quad S1 \qquad (2.4.15a)$$

$$= u_{ij} [\partial_c^2 u]_{ij} \quad : \quad S2 \text{ and } S3 \qquad (2.4.15b)$$

2.4.4. Iterative Techniques

The discretized approximation to the integral equation may be summarized as follows:

$$u_{ij} = u_{Lij} + \delta u_{Wij} + u_{Fij} \qquad (2.4.16)$$

$$u_{Fij} = \sum_k \sum_l a_{Fij}^{kl} \bar{g}_{kl} \qquad (2.4.17)$$

$$u_{Wij} = \sum_k a_{Wij}^k u_{kl}, \quad l = l_{max} \qquad (2.4.18)$$

The field source term $\bar{g}$ may be calculated from anyone of Eqs. (2.4.13), depending upon the scheme chosen. At the shock point, $\bar{g}$ is obtained using either Jameson's shock operator (referred to as JA) from (2.4.13) or the elliptic operator (referred to as EL) from Eq. (2.4.15).

Equation (2.4.16) represents a system of non-linear algebraic equations which must be solved iteratively. Numerous iterative schemes are available for the solution of such systems [94]. In the present case, this system of equations is solved by a Direct Iteration Scheme (DIS) as proposed here or by a Quasi-Newton Scheme (QNS) as was done by Piers and Slooff [73]. The latter scheme, with a super-linear convergence rate, is generally faster than the direct scheme but is found to be unstable for the wall interference case. The direct scheme, on the other hand, is found to be quite inefficient for free air calculations.

For the free air case, either $\bar{g}$ or u may be treated as the iterative unknown. For the wall interference case however, only u is suitable as the iterative unknown because the wall integral requires the value of u for its evaluation. Further, numerical experiments show that it is desirable to add terms representing artificial time derivatives of u and u_x to the iterative formulation of Eqs. (2.4.16) to improve the stability and convergence properties of the scheme. Assuming that an initial estimate $\bar{g}^{(n)}$ is available at the n-th iteration, Eq. (2.4.16) may be rewritten in the following form for the free air case.

$$u_{ij}^{(n)} = u_{Lij} + u_{Fij}^{(n)} \tag{2.4.19}$$

$$R_{ij}^{(n)} = \bar{g}_{ij}^{(n)} - \bar{g}_{ij}^* - \alpha_1 (u_{ij}^{(n)} - u_{ij}^{(n-1)}) - \alpha_2 (u_{x_{ij}}^{(n)} - u_{x_{ij}}^{(n-1)}) \tag{2.4.20}$$

Here $\bar{g}^*$ is obtained from Eq. (2.4.13) using the estimates $u^{(n)}$ and $u_x^{(n)}$. α_1 and α_2 are iteration parameters used to control the magnitude of the time damping terms and hence the stability and convergence rate of the iterative scheme. The next iterate is obtained from

$$\bar{g}_{ij}^{(n+1)} = \bar{g}_{ij}^{(n)} + \omega\Delta\,\bar{g}_{ij}^{(n)} \tag{2.4.21}$$

where ω is a damping factor for the correction vector. The value of ω is usually less than unity and is different at subsonic and supersonic points.

The correction vector $\Delta\bar{g}^{(n)}$ is obtained by solving a system of linear equations in the quasi-Newton scheme. In the direct iteration scheme, it is simply the negative of the residual. Thus

Quasi-Newton Scheme: $\quad C_{ij}^{kl(n)}\,\Delta\bar{g}_{kl}^{(n)} = -R_{ij}^{(n)} \tag{2.4.22}$

Direct Iteration Scheme: $\Delta\bar{g}_{ij}^{(n)} = -R_{ij}^{(n)} \tag{2.4.23}$

The matrix C is computed from

$$C_{ij}^{kl(n)} = \frac{\partial R_{ij}^{(n)}}{\partial\bar{g}_{kl}^{(n)}} \tag{2.4.24}$$

The matrix C is approximated by its tri-diagonal elements and the correction vector is easily determined by an explicit inversion formula.

For the wall case, assuming an initial estimate $u^{(n)}$ we have,

$$R_{ij}^{(n)} = u_{ij}^{(n)} - u_{Lij} - u_{Wij}^{(n)} - u_{Fij}^{(n)} + \alpha_1(u_{ij}^{(n)} - u_{ij}^{(n-1)})$$

$$+ \alpha_2(u_{xij}^{(n)} - u_{xij}^{(n-1)}) \qquad\qquad (2.4.25)$$

The next iterate is given by

$$u_{ij}^{(n+1)} = u_{ij}^{(n)} + \omega\Delta\, u_{ij}^{(n)}$$

The correction vector is calculated by a procedure similar to that outlined for $\Delta\bar{g}$ in the free air case.

2.5. RESULTS AND DISCUSSION

The numerical schemes were coded in FORTRAN and run on the DEC 1090 computer. Most of the calculations were carried out interactively on the time-sharing mode. In the following sections, results obtained for axisymmetric flow past slender, general parabolic bodies of resolution in free air as well as in the presence of perforated wind tunnel walls will be presented.

The equation for the general parabolic body of revolution is given in Appendix C. The linear solution u_L, given by Eq. (2.4.5), was computed numerically, evaluating the integral by Simpson's rule. For the case of a body with maximum thickness at the centre, the numerically computed values were compared with exact values obtained by evaluating the integral in Eq. (2.4.5) analytically. The two results were found to be in agreement within an error margin of about 0.5 percent. The linear solution was then used as an input for the iterative solution of the non-linear problem.

Finally it is remarked that the value of the artificial viscosity parameter λ was set equal to unity in all the following calculations except when stated otherwise.

2.5.1. Grid Effects

Following normal practice, the grid related parameters for obtaining satisfactory solutions were established by numerical experiments. These results are discussed first.

Figures 2.3(a to e) present the effects of the grid related variables on the surface pressure distribution for a parabolic body of revolution without sting and with maximum thickness section located at the mid-point of the axis. The free-stream Mach number and body fineness ratio chosen to study these effects were 0.98 and 10 respectively. The numerical calculations were carried out for a typical free air case using scheme S_1 and shock operator JA.

Figures 2.3a and b show respectively the effect of axial and lateral grid extents. For the three grids used in each of these figures (namely $j \times i = 10 \times 40$, 10×35 and 10×31 in Figure 2.3a and 10×35, 7×35 and 6×35 in Figure 2.3b), it is seen that the surface pressure distribution obtained on the smallest grid differs hardly from that obtained on the largest. Hence it is concluded that a computational domain bounded by $-0.20 \leq x \leq 1.20$ and $0 \leq r \leq 0.35$ is sufficient to obtain satisfactory results on the body surface. This may be compared with the axial grid extent of $-2.36 \leq x \leq 3.36$ used by Bailey [84] in his finite-difference calculations. For $M_\infty =$
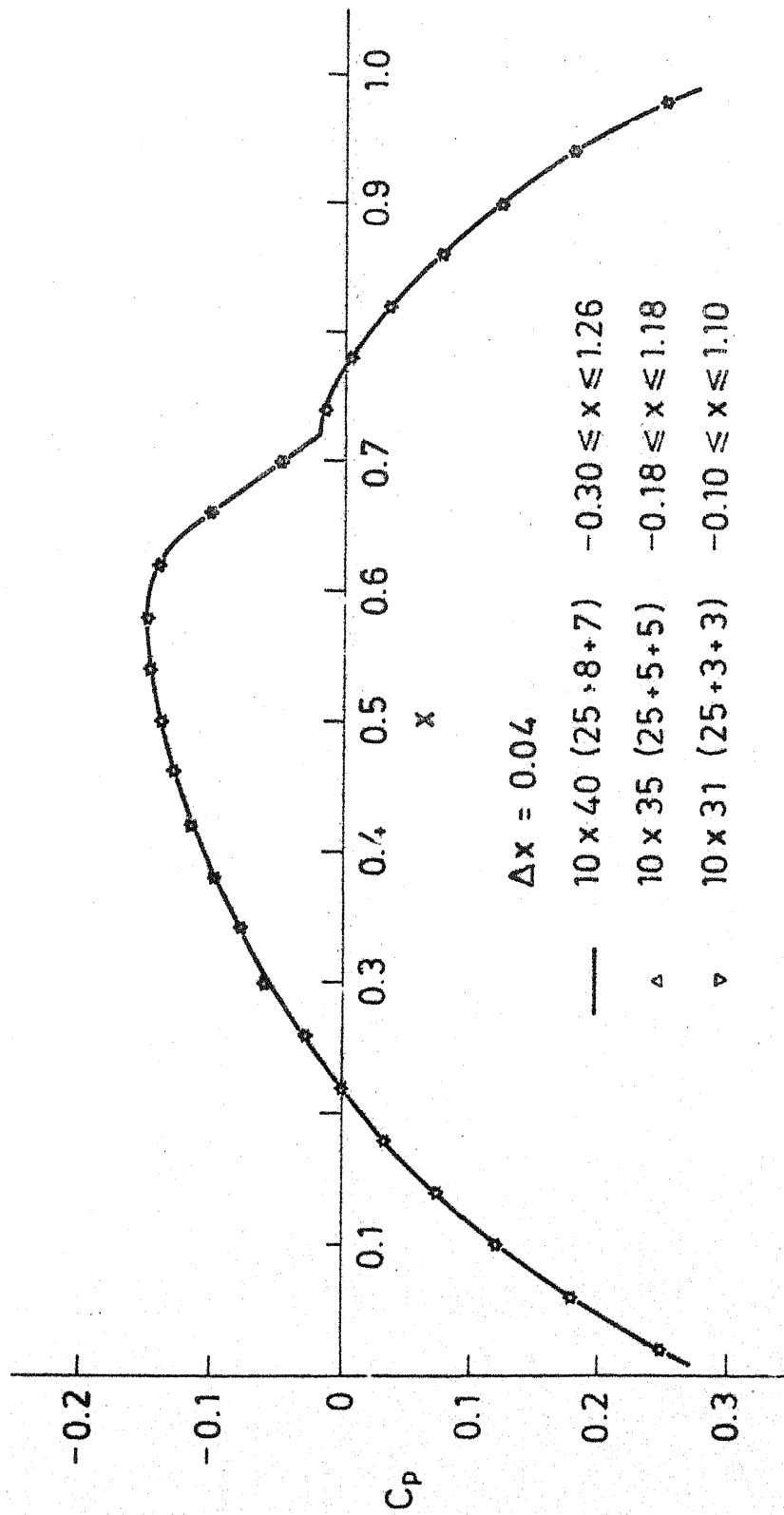
Fig. 2.3a.   Effect of Axial Grid Extent on the Surface Pressure Distribution.
$M_\infty = 0.98$, FR = 10 , Free Air , No Sting
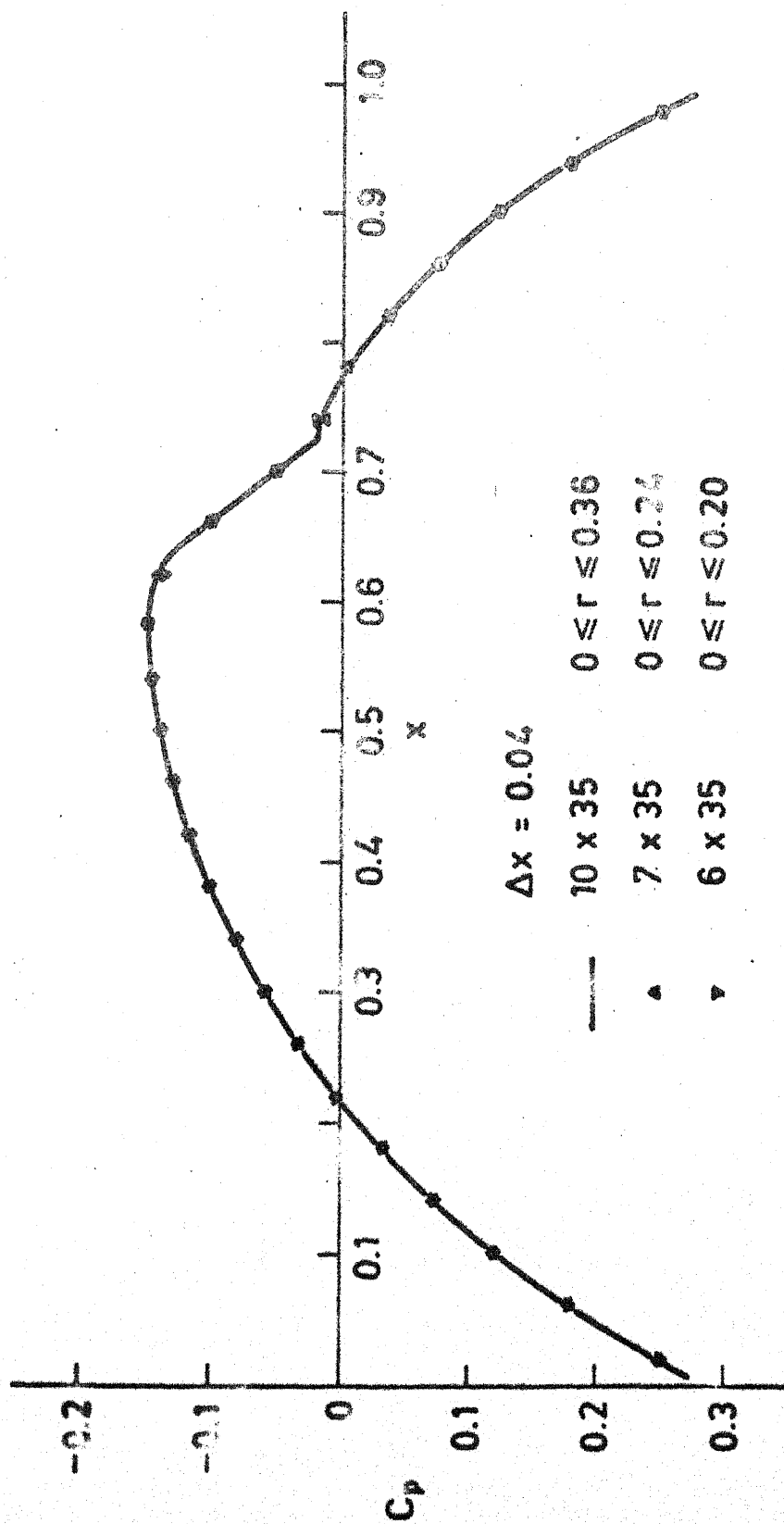
Fig. 2.3b. Effect of Lateral Grid Extent on the Surface Pressure Distribution. $M_\infty = 0.96$, FR = 10, Free Air, No Sting
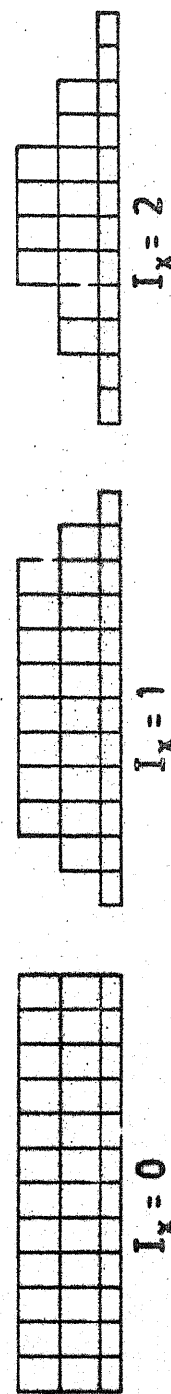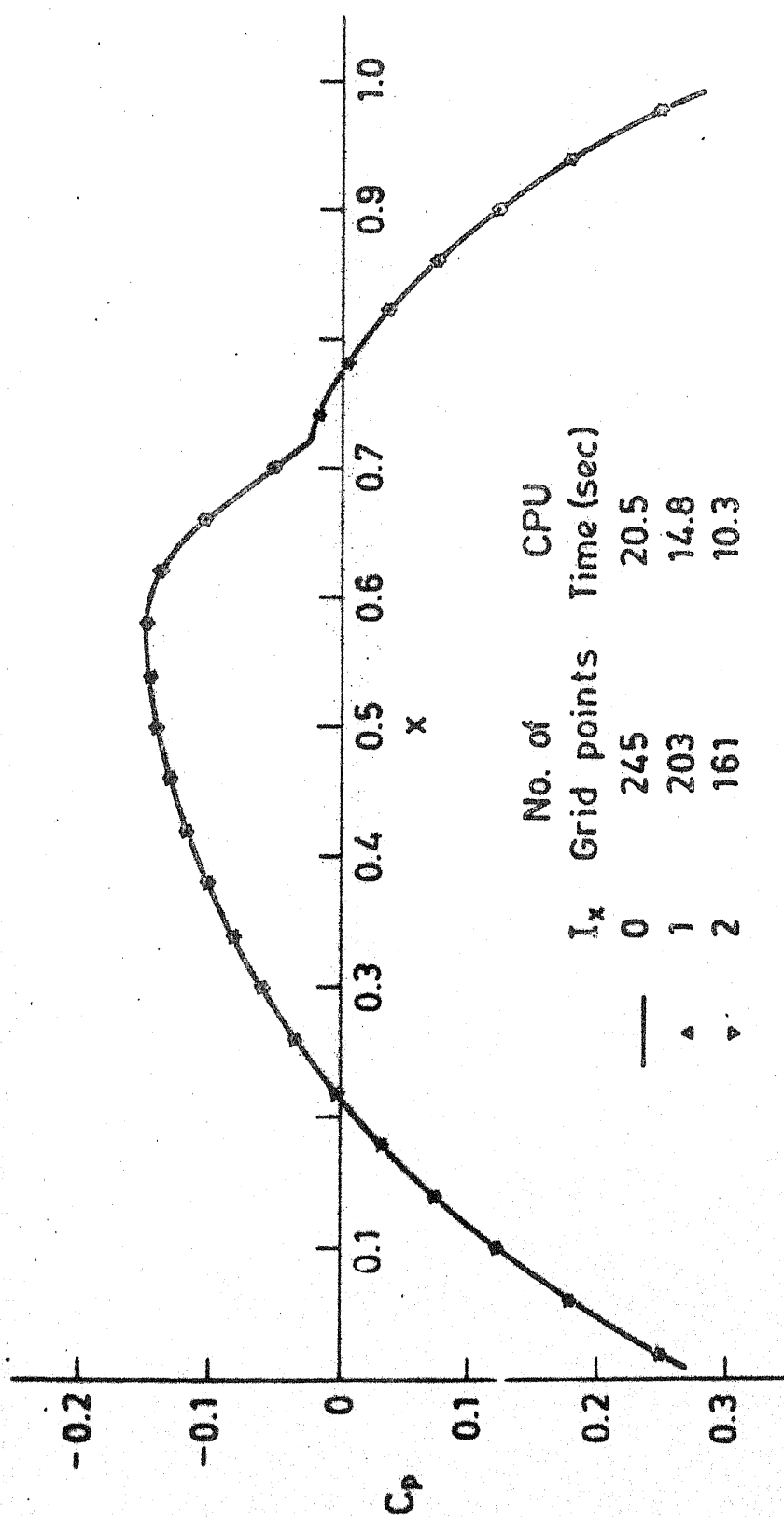
Fig. 2.3c. Effect of Grid Extent on Surface Pressure Distribution. $M_\infty = 0.98$, FR = 10 Free Air, No Sting

56
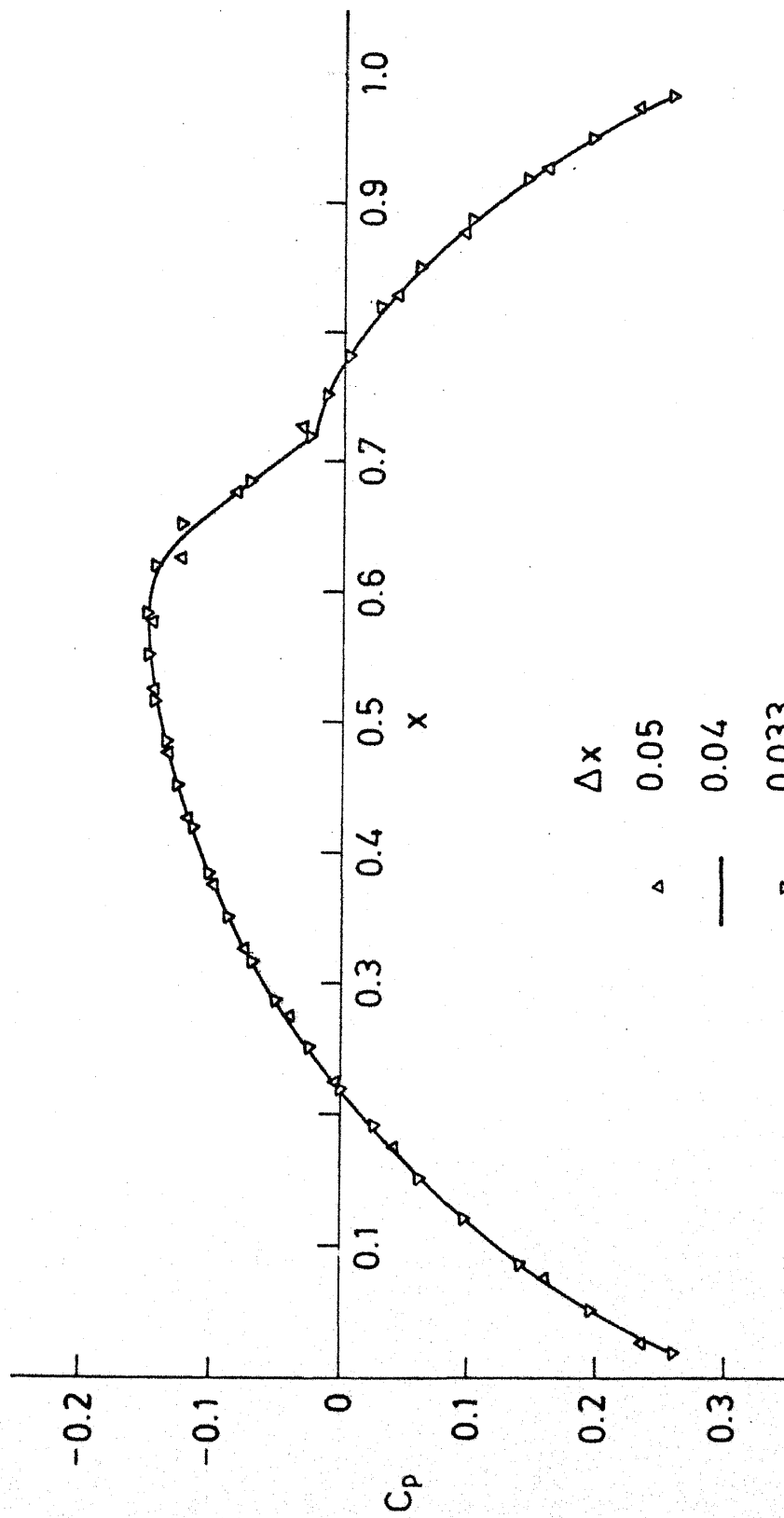
Fig. 2.3d. Effect of Mesh Size on Surface Pressure Distribution. $M_\infty = 0.98$, FR = 10, Free Air, No Sting
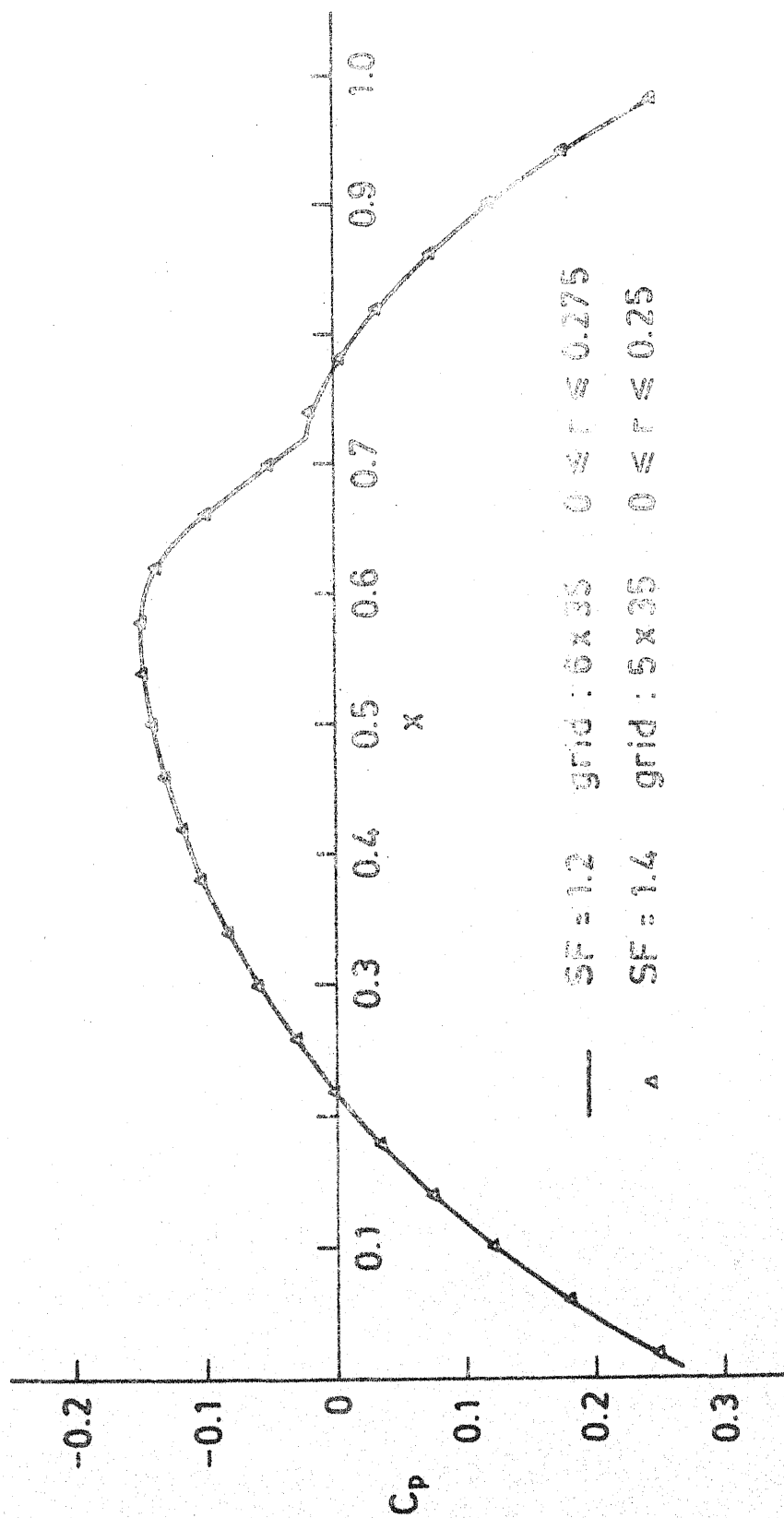
Fig. 2.3e. Effect of Vertical Mesh Stretching on the Surface Pressure Distribution.
$M_\infty = 0.98$, FR = 10, Free Air, No Sting

0.99, the above range of r on the transformed plane corresponds to $0 \leq \bar{r} \leq 2.5$ on the physical plane.

Further economy of grid points was achieved by progressively reducing the number of streamwise grid points at every j-level, moving away from the body axis (see Figure 2.3c). The figure shows that with $I_x = 2$, the number of grid points could be reduced to 161 from 245, leaving the computed pressure distribution on the surface unaffected. The CPU time gets nearly halved with their reduction in the number of grid points. Calculations presented here were however carried out with $I_x = 1$ unless stated otherwise. Further it is pointed out that for the wall interference calculations all the grid points on the wall boundary are esential for the evaluation of the wall integral and hence $I_x = 0$ for all these calculations.

Figure 2.3d shows the effect of mesh size on the surface pressure distribution. As expected, a coarse mesh causes a slight smearing of the shock. This is evident from Figure 2.3d for $\Delta x = 0.05$. Figure 2.3e shows the effect of a stretching factor in the r-direction on the surface pressure distribution. This factor increases the aspect ratio of the rectangular cells in the r-direction as one moves away from the body axis. Thus for a fixed lateral grid extent fewer number of j-levels are required than if a square mesh were employed. For the two values of SF = 1.2 and 1.4 used in Figure 2.3e, the computed pressure distributions turn out to be identical. It might seem that still larger values for the stretching factor could be used to reduce the number of grid points further, but since the iteration parameters ω, α_1 and α_2 are related to mesh aspect

ratio, a larger SF could, and often did, lead to divergence of the iteration scheme. For example with $M_\infty = 0.99$ and FR = 10, it was not possible to obtain convergence for the free air case with SF ⩾ 1.0, whatever the values of ω, α_1 and α_2 used. The values of SF used in the present calculations generally varied from 1 to 1.2 depending upon the free stream Mach number and body fineness ratio, the smaller value being used for the stronger super-critical cases.

2.5.2. Comparison of the Results of the Schemes S1, S2 and S3 and Shock Operators JA and EL

Figures 2.4 and 2.5 present a comparison of the computed surface pressure distributions for a parabolic body of resolution without sting in free air, using the schemes S1, S2 and S3. Results obtained using both the Jameson (JA) and the elliptic (EL) shock operator are presented in the figures. Figure 2.4 displays the pressure plots for $M_\infty = 0.99$, FR = 10 whereas Figure 2.5 shows the same plots for $M_\infty = 0.98$, FR = 6.

Considering Figure 2.4a, we see that scheme S2 shows an over expansion just ahead of the shock while elsewhere, the three results are in close agreement with one another. It is noted however that scheme S1 needs just about half as many iterations to converge as do the other two schemes. The shock operator used in this case was JA. Figure 2.4b shows the results of the same calculation using the EL shock operator. The scheme S3 in this case went into an oscillation near convergence. Schemes S1 and S2 converged in about 50 iterations. Here again the pressure distribution obtained from scheme S2 is seen to overshoot just before the shock.
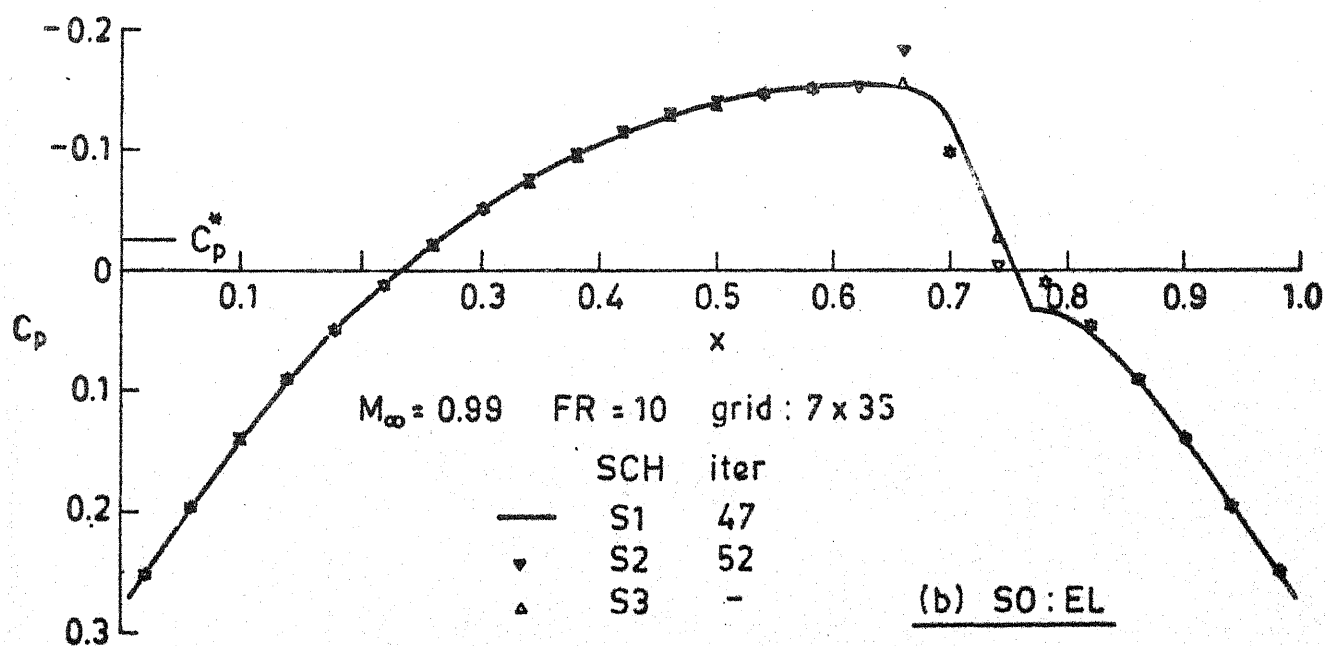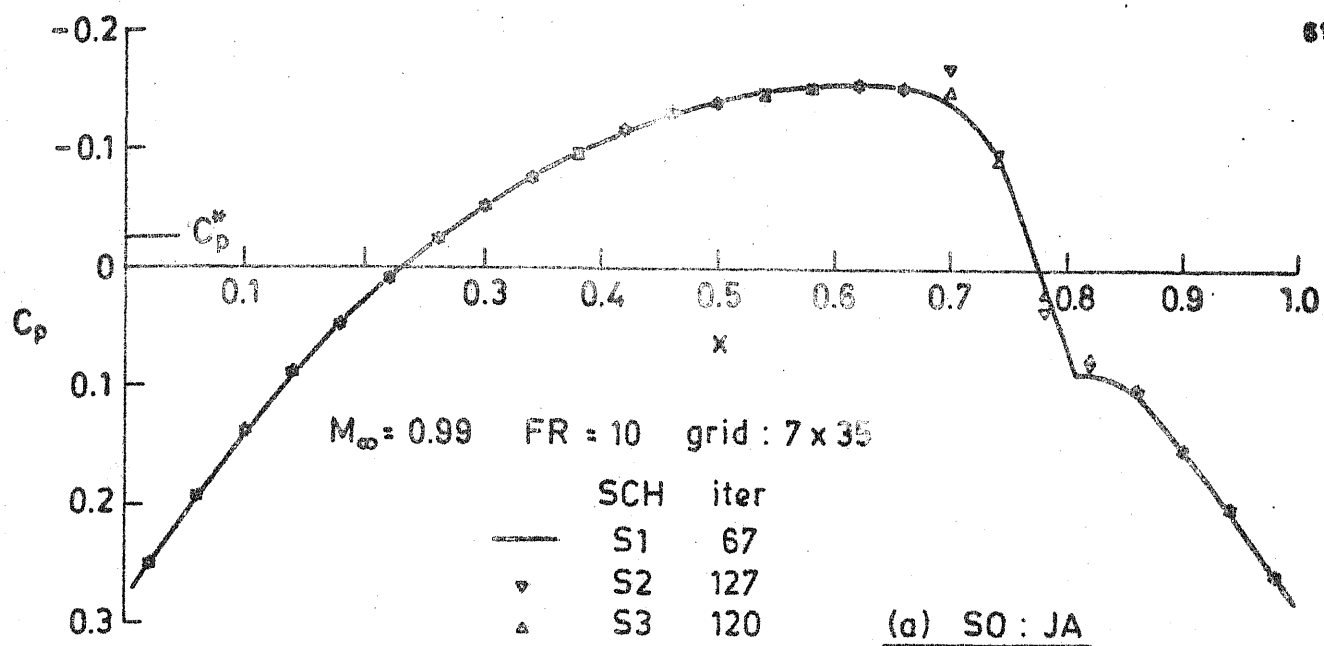
Fig. 2.4. Comparison of the Schemes S1, S2 and S3 : Surface Pressure
Distribution on a Parabolic Body of Revolution in Free Air.
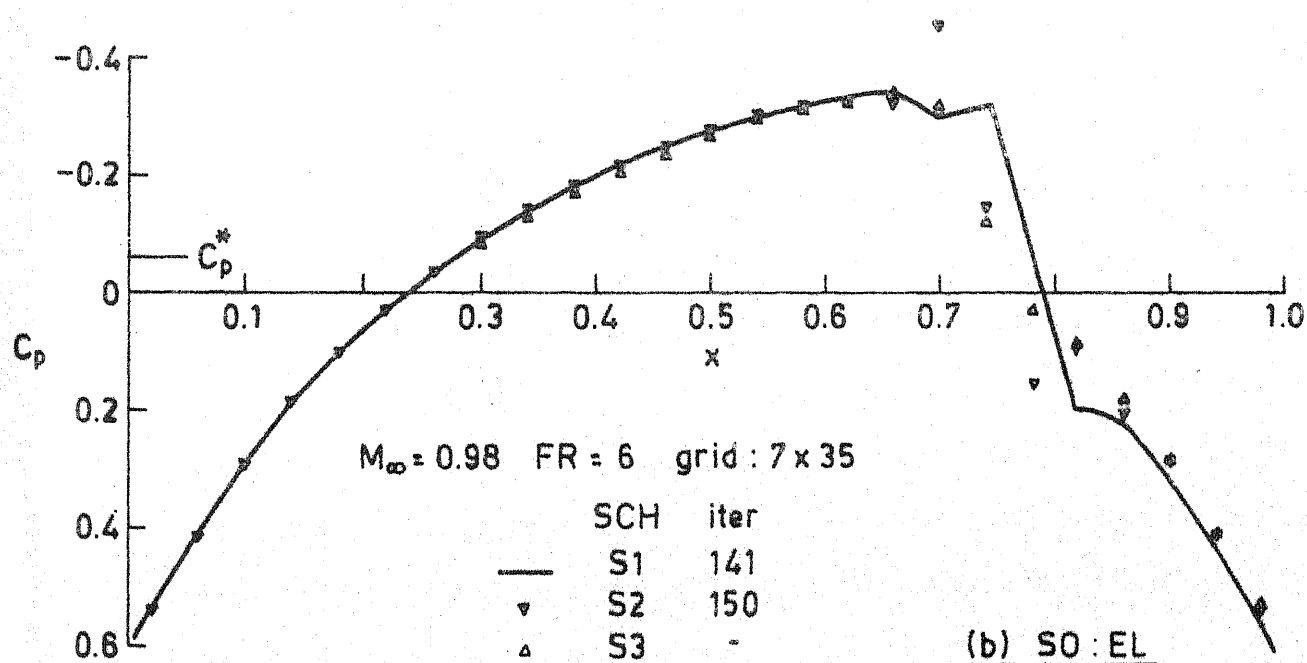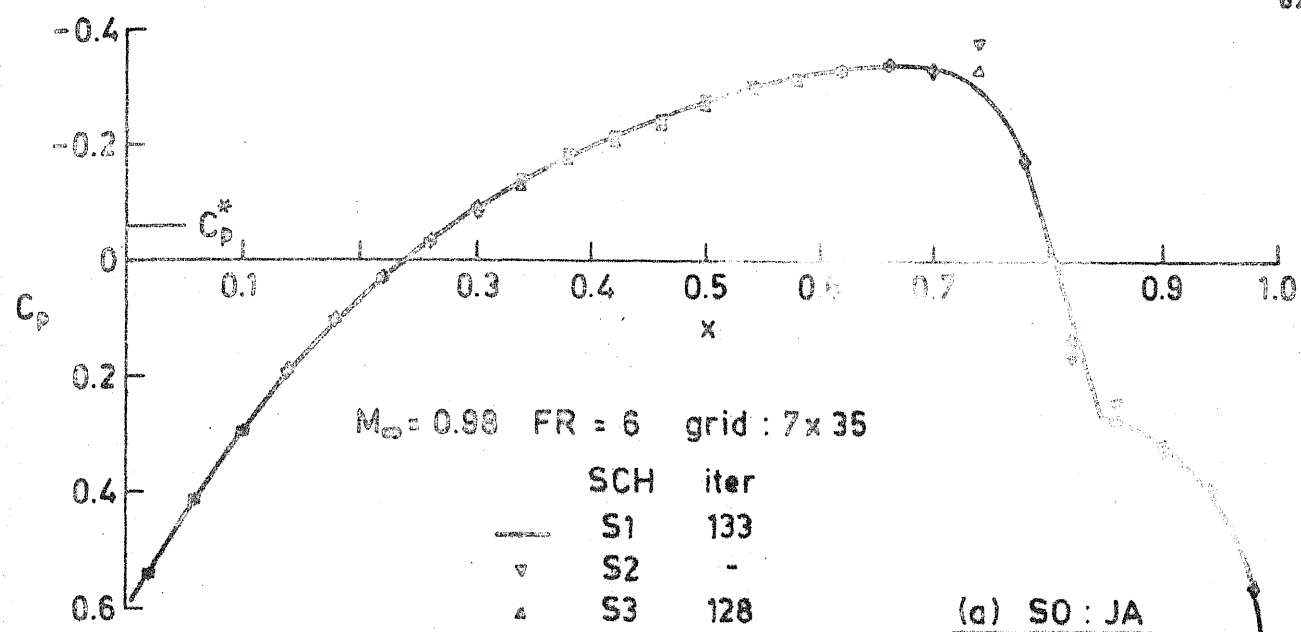
Fig. 2.5.   Comparison of the Schemes S1, S2 and S3 : Surface Pressure
Distribution on a Parabolic Body of Revolution in Free Air.

Figure 2.5 shows the results of the above calculations repeated with $M_\infty = 0.98$ and FR = 6. Comments made in relation to Figure 2.4 apply almost identically to the results presented in Figure 2.5. In addition it is noted that in Figure 2.5b, which shows the pressure distribution obtained with an elliptic operator, scheme S1 also results in a slight over-expansion just ahead of the shock. The significant difference in the shock location calculated by scheme S1 and that of S2 and S3 for this case is due to the occurrence of this over-expansion. It was thought that such an over-expansion occurred because of insufficient dissipation. The calculations were therefore repeated for higher values of λ but it was found that the non-uniformity persisted and the shock location remained unchanged. Here again it is noted that scheme S2 in Figure 2.5a and S3 in Figure 2.5b went into oscillations near convergence.

In all these cases, the surface pressure distribution shows a smooth development over the subsonic region ahead of the shock, including the region around the sonic point. The subsonic portion is, as expected, insensitive to the scheme used. There is also a small region of smooth deceleration, instead of an abrupt transition from the point of the velocity peak to the beginning of what can be regarded as the shock transition region.

Figures 2.6a and b show comparisons of the surface pressure distributions obtained on the same body in free air, using the elliptic and Jameson shock operators, for $M_\infty = 0.99$, FR = 10 and $M_\infty = 0.98$ and FR = 6 respectively. The S1 scheme was used for these calculations. In both cases it is seen that
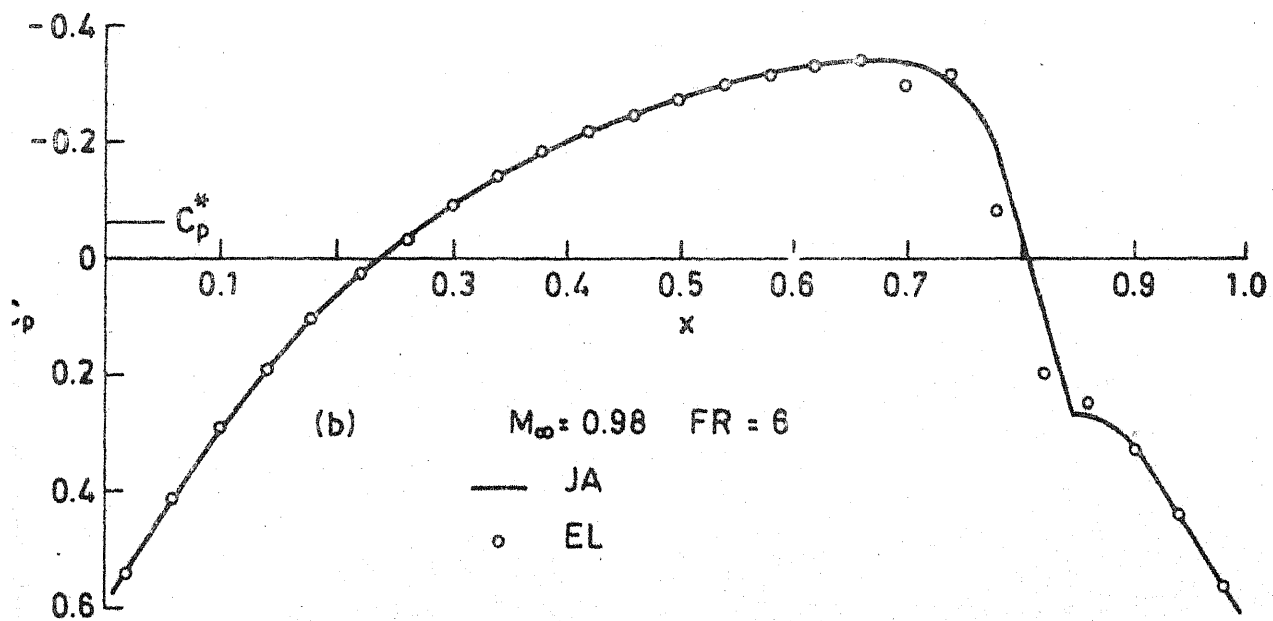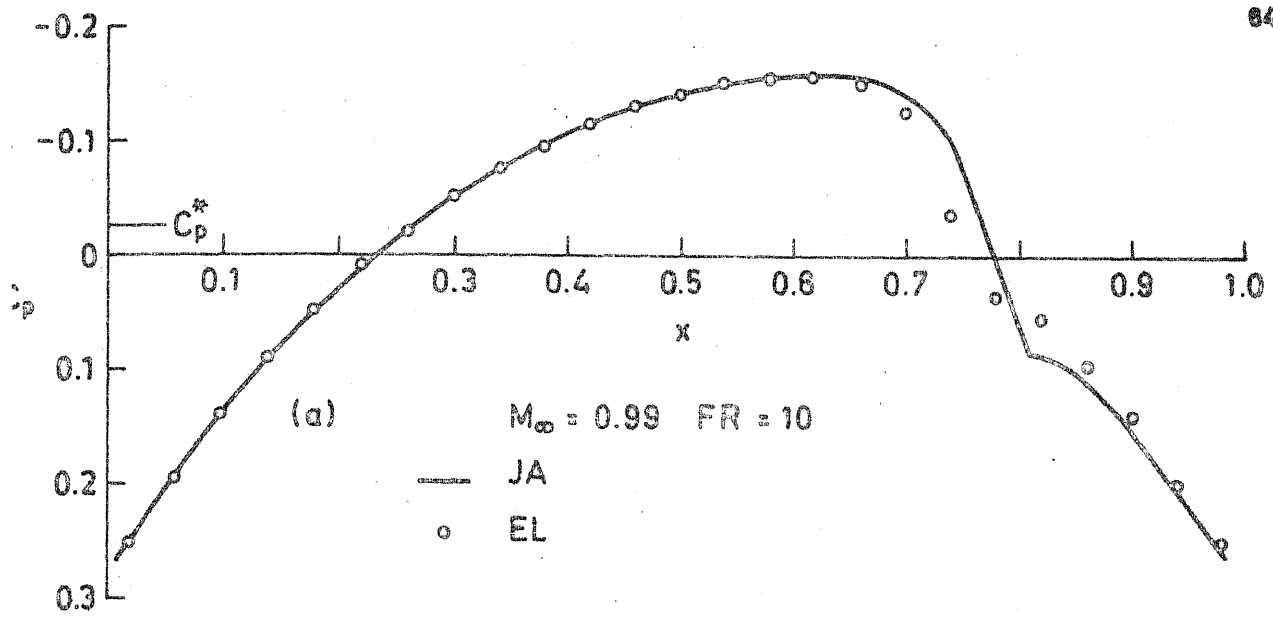
Fig. 2.6. Comparison of the Shock Operators (SO) : Surface Pressure
Distribution over a Parabolic Body of Revolution. SCH : S1,
Free Air, grid : 7 x 35 .

the elliptic shock operator produces a weaker shock, located about half a mesh width ahead of that predicted by the Jameson operator. For weaker super-critical flows however, this effect does not emerge clearly since in such cases, the present method tends to smear out the captured shocks.

From the above discussion it is clear that scheme S1, together with the Jameson or elliptic shock operator is the best choice from the point of view of computing time, as well as, obtaining satisfactory solutions. Results of further calculations presented here were therefore obtained using the S1 scheme together with either the Jameson (JA) or elliptic (EL) shock operator.

2.5.3. Comparison of Present Results with Other Experimental and Numerical Results

A comparison of the results of the present computations with other theoretical and experimental results is presented in this section.

(a) Free Air Results

Figure 2.7 shows the computed results in free air for a parabolic body of revolution with sting at x = 0.854, FR = 10 and at a free-stream Mach number 0.975, using the EL operator. Pressure distributions on the body surface and at four radial locations away from the body are compared with the experimental results of McDevitt and Taylor [95]. The agreement is seen to be reasonably good. Figure 2.8 shows the pressure distributions obtained on general parabolic bodies of revolution with axial locations of maximum body thickness at 0.3, 0.4, 0.6 and 0.7
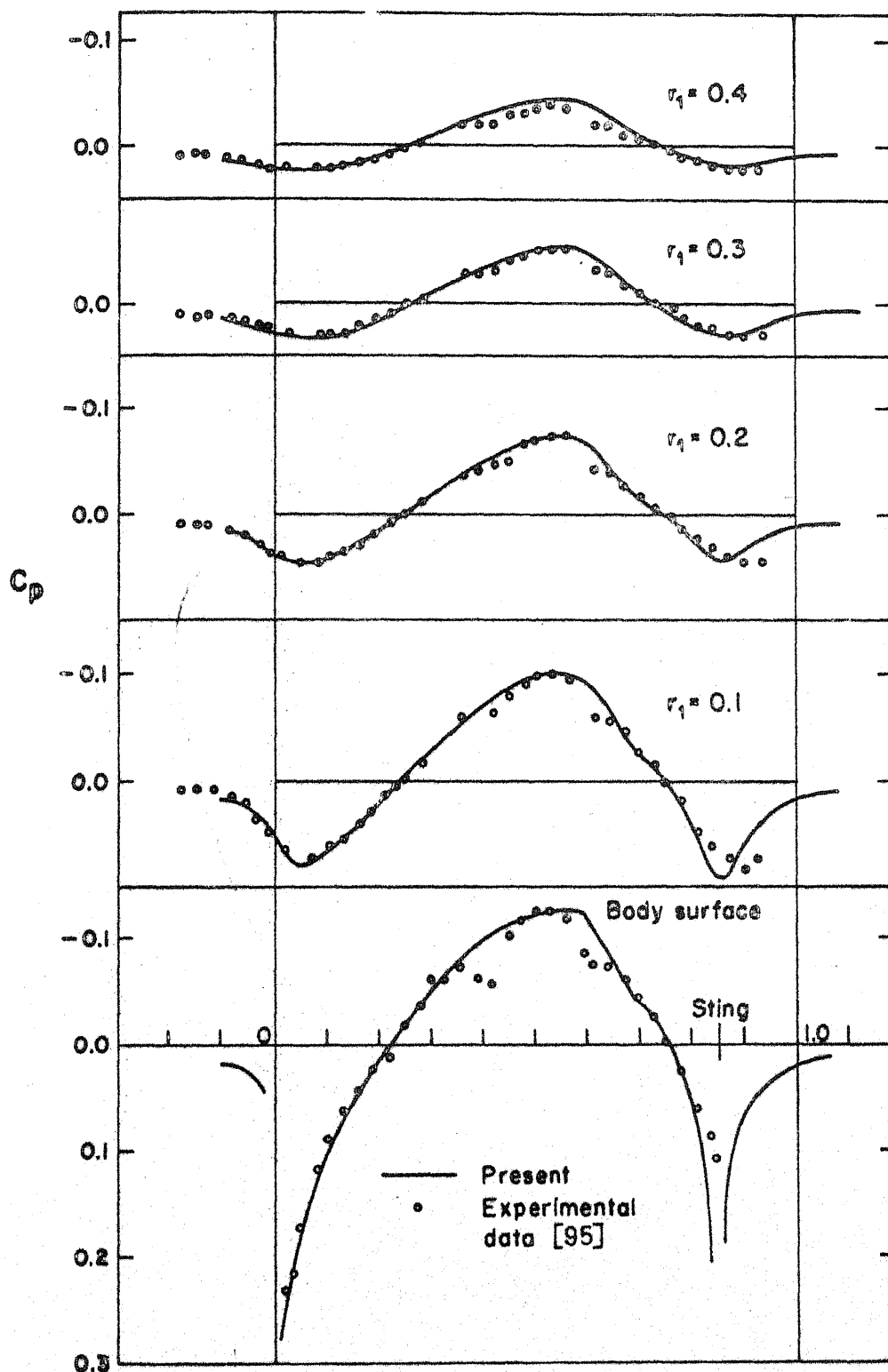
Fig. 2.7. Comparison of Pressure Distribution for Flow over a Parabolic Body of Revolution at Different Radial Locations with Experimental Data. $M_\infty = 0.975$, FR = 10, SCH : S1, SO : EL, Free Air.
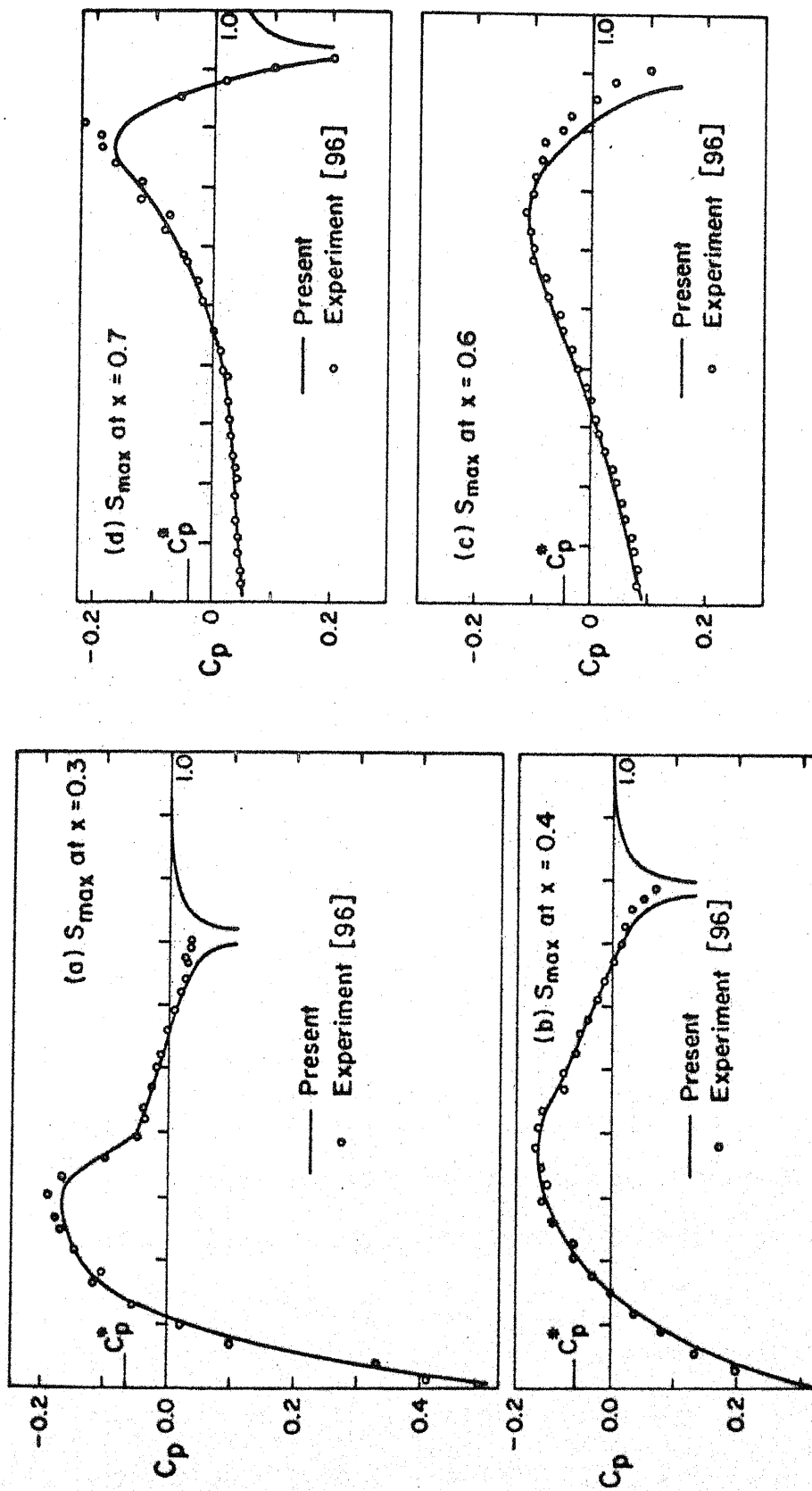
Fig. 2.8. Comparison of Computed Pressure Distributions with Experimental Data of McDevitt and Taylor for General Parabolic Body of Revolution. $M_\infty = 0.975$, $FR = 12$, $SCH:S1$, $SO:EL$, Free Air.

and FR = 12. The free-stream Mach number is 0.975. The agreement with experimental data [96] is again in general satisfactory. The discrepancy near the body sting may be caused by the failure of small disturbance theory in this region. Use of Jameson shock operator gave almost the same results for these cases since the shocks are comparatively weak and smeared for all of them.

Figures 2.9a and b show comparison of the present results with results of finite-difference calculations. Present results, obtained using the JA shock operator, are plotted for a body with sting in free air. In Figure 2.9a with $M_\infty = 0.99$ and FR = 10, it is seen that the non-conservative shock location from Bailey's [84] finite-difference calculations, is about half a mesh width ahead of the location predicted by the present quasi-conservative calculations. The pressure distribution obtained by using the EL operator for this case is plotted in the same figure and is seen to be closer to Bailey's results. Figure 2.9b shows the results of the same calculation for $M_\infty = 0.98$ and FR = 6. This is compared with Krupp's [97], non-conservative finite-difference calculations. As in the previous case, here too the shock location predicted by Krupp's scheme is about half a mesh width ahead of the present shock. In both figures it is noted that the surface pressure distribution calculated by the present scheme, in the fore region of the body is somewhat flatter compared to the finite-difference results. But for this discrepancy, the results of the present calculations are in fairly good agreement with experiment as well as with finite-difference results.
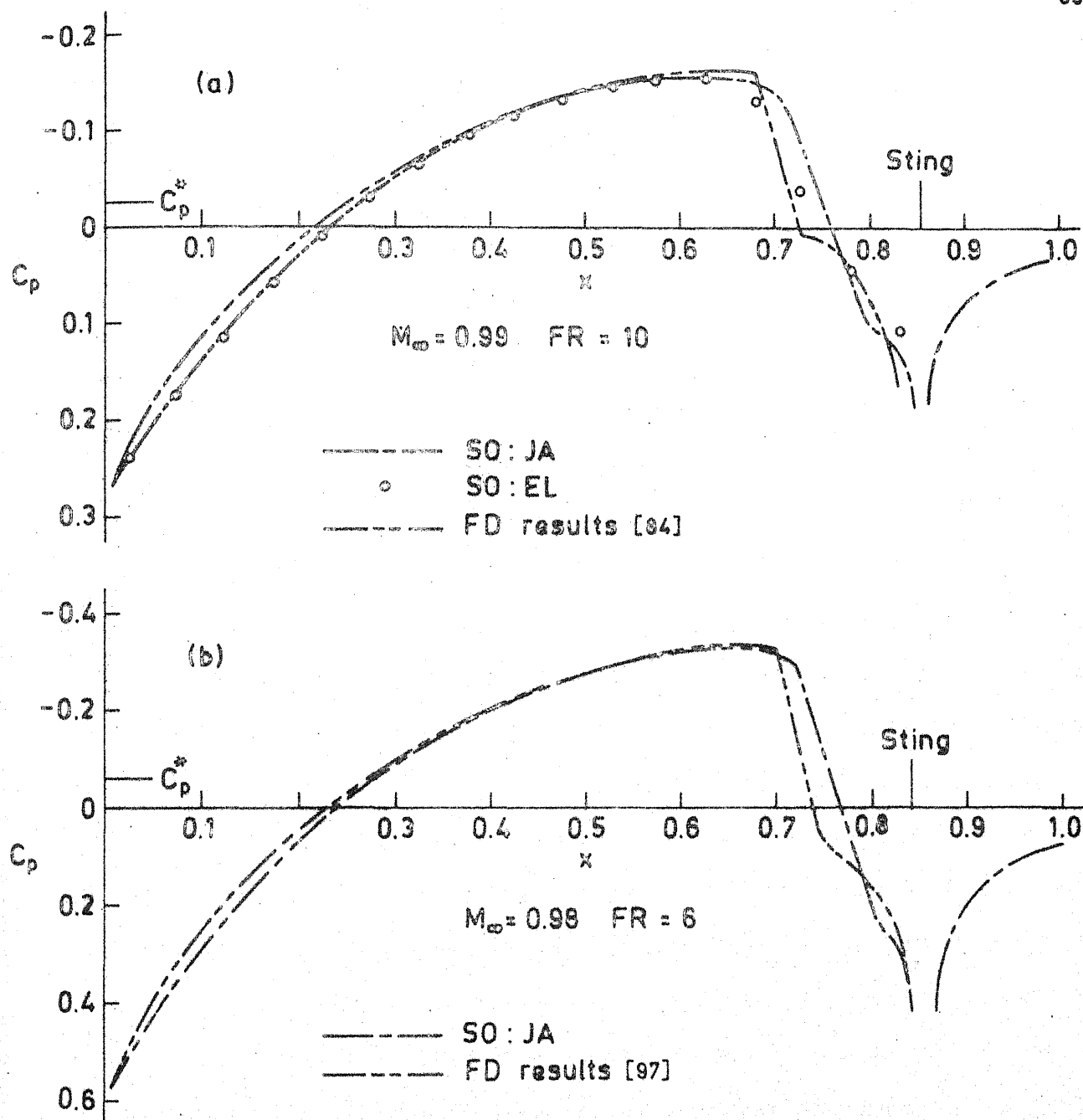
Fig. 2.9. Comparison of Computed Pressure Distribution for Parabolic
Body of Revolution with Finite-Difference Results, Free Air.

(b) Wall Interference Results

Figures 2.10 and 2.11 show, from calculations using an EL operator, the effect of the presence of a perforated wall on the surface pressure distribution for bodies of two different fineness ratios at two different free stream Mach numbers.

In general it is seen that the presence of the wall has a significant effect on the computed surface pressure distributions. For the weaker super-critical cases, with $M_\infty = 0.975$, FR = 10 or $M_\infty = 0.94$, FR = 6 (Figures 2.10a and 2.11a), the effect is more pronounced in the shock region and relatively small over the subsonic and accelerating supersonic regions of the flow. For the stronger super-critical cases with $M_\infty = 0.99$, FR = 10 and $M_\infty = 0.98$ and FR = 6, the surface pressure distribution changes over a greater region including the accelerating supersonic portion and the region of shock transition, Figures 2.10b and 2.11b. It is seen that increasing porosity causes the shock to move upstream by about 2 percent of the chord. For smaller free-stream Mach numbers, with fixed FR, there is a tendency for the shock to get smeared with increasing porosity. In general the surface velocities attained are smaller as the porosity factor increases. This makes physical sense because if part of the mass flow is bled away, the velocities must be smaller. The solid wall case with p = 0 gives the maximum surface velocities. However convergence could not be achieved for this case for the stronger super-critical cases (for example, $M_\infty = 0.99$, FR = 10).
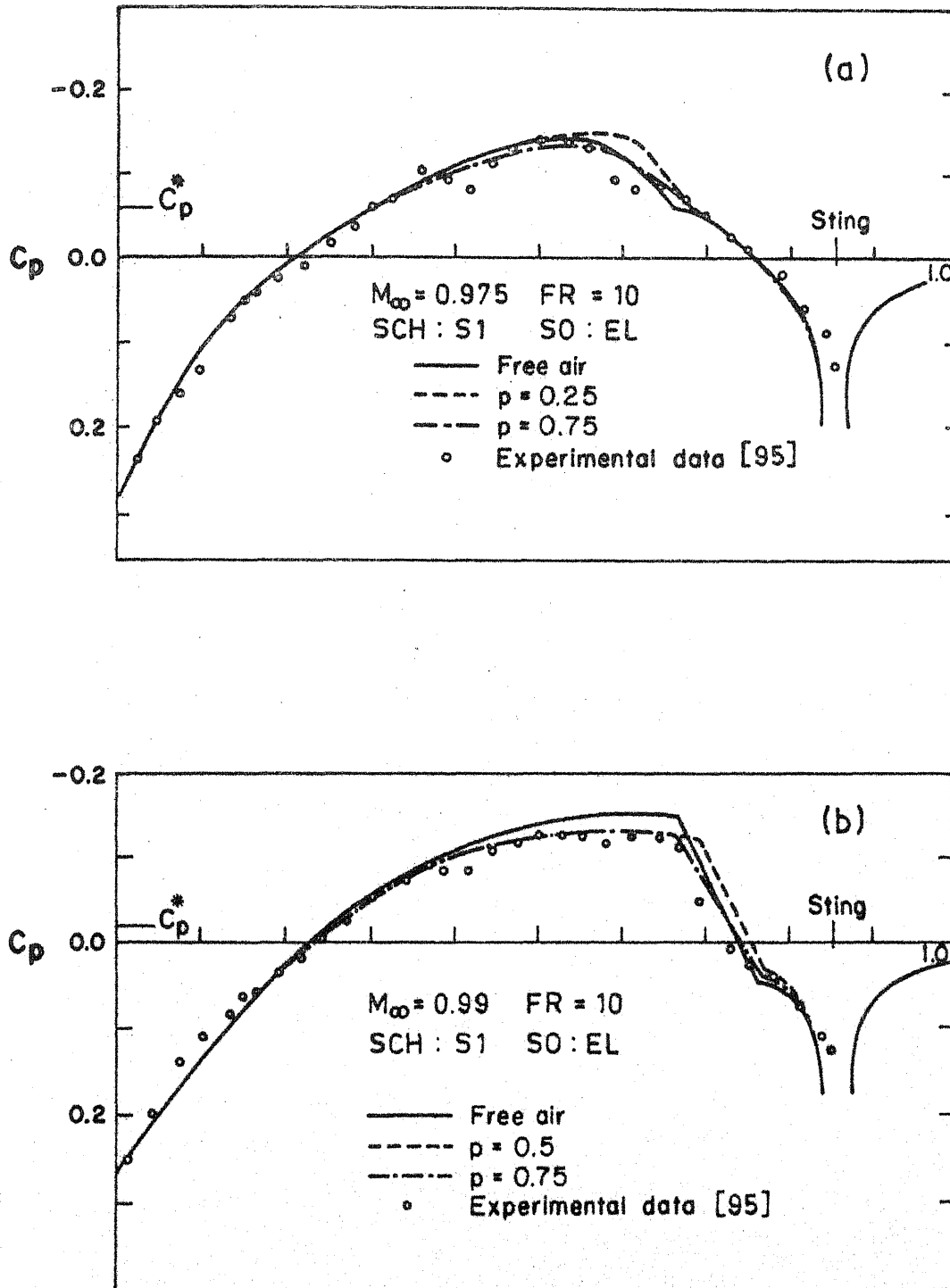
Fig. 2.10. Effect of Porous Wall Boundary on the Surface Pressure Distribution for a Parabolic Body of Revolution with Sting.
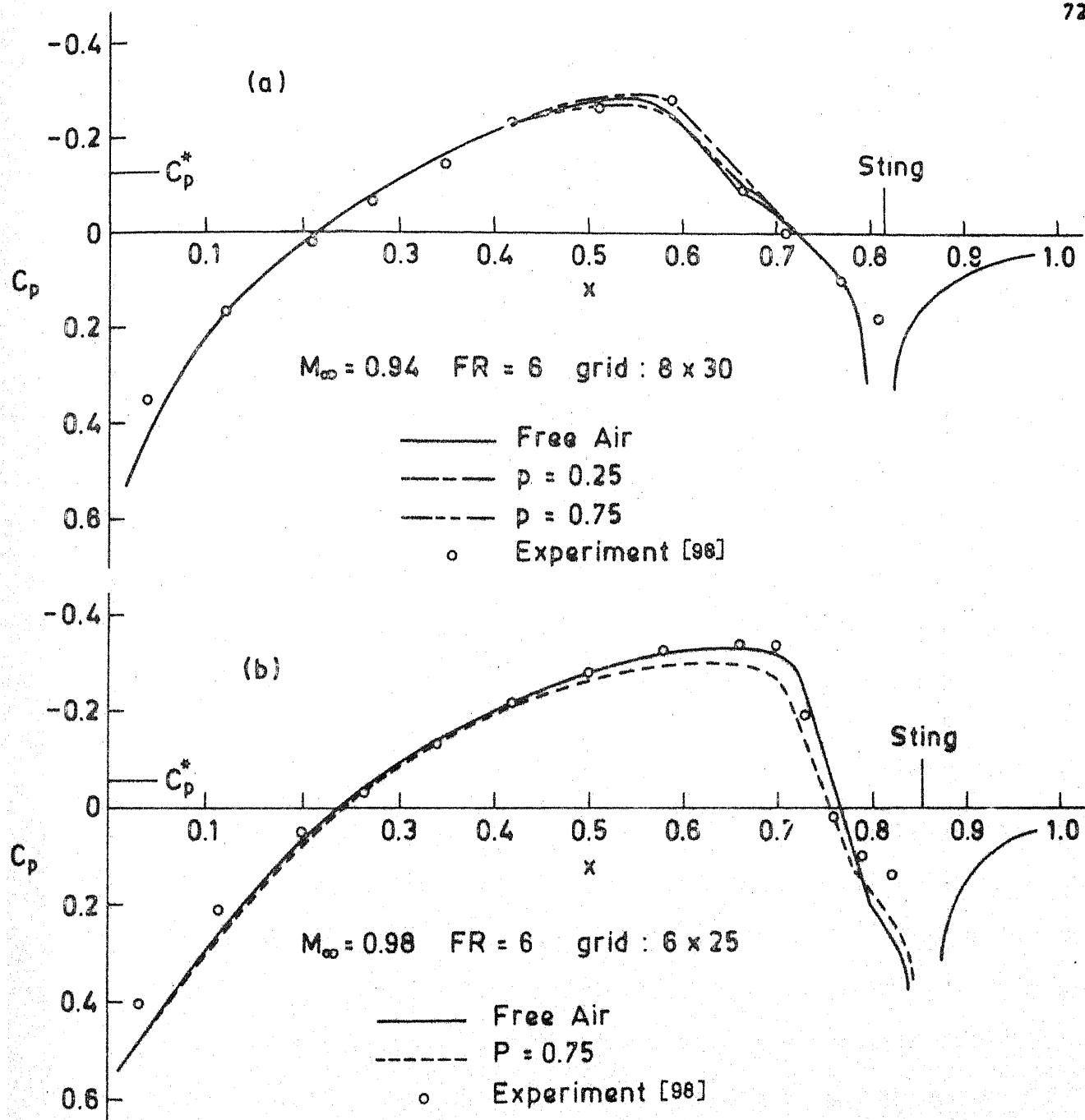
Fig. 2.11. Effect of Porous Wall Boundary on the Pressure Distribution over a Parabolic Body of Revolution. SCH : S1, SO : EL.

Figures 2.10 and 2.11 also show the experimental data obtained for these cases plotted along with the present theoretical results. The comparisons shown by Figures 2.10a and b suggest that the experimental shock may be approached by increasing porosity.

Figures 2.11a and b show the results of the present calculations compared with experimental data [98] for a body with FR = 6. Here the agreement with free air results seems to be better than that with wall interference. As for the shock, the experimental shock strength seems to be in good agreement with the calculated value for $M_\infty = 0.94$ whereas for $M_\infty = 0.98$, the calculated shock is somewhat stronger. The shock locations in both cases are however in agreement with experiment.

2.5.4. Convergence Characteristics of the Iteration Scheme

As mentioned in Section 2.4.4, free air calculations were done by iterating on $\bar{g}$ and wall interference calculations by iterating on u. The Quasi-Newton Scheme (QNS) or the Direct Iteration Scheme (DIS) could be employed for both. QNS proved to be more efficient for the free air case whereas DIS was found superior for the wall interference calculations.

Figure 2.12 shows, for a fixed ω, the effect of the iteration parameters α_1 and α_2 on the convergence history for the QNS. These calculations were done in free air for a body with no sting, with FR = 10 and $M_\infty = 0.99$. The value for ω is given in pairs, the first value being used at subsonic points and the second at supersonic points. Since we had no apriori estimates for ω, α_1 and α_2, the solution for a prescribed free

stream Mach number and thickness was attempted by setting

$\alpha_1 = \alpha_2 = 0$. ω was set at some value usually less than unity,

the value used at supersonic points being 0.1 less than that at

sub-sonic points. A larger value for ω generally implies

faster convergence but could also lead to divergence depending

upon the free stream Mach number and thickness.

Figure 2.12 shows, for the QNS, $||R||$ of the residual

plotted against the iteration number. The two curves demons-

trate the effectiveness of the artificial time-dependent terms

in achieving faster convergence. The number of iterations

needed with $\alpha_1 = 7.0$ and $\alpha_2 = 0.05$ is 48 whereas with $\alpha_1 = \alpha_2 = 0$,

76 iteration were needed for the same. For this case $||R||$

showed a more or less uniform decreasing behaviour from the

maximum value at the beginning of the iteration to its value at

convergence. The oscillatory behaviour noticed for the plane,

two-dimensional case (see Section 2.5.5) and in the wall case

below, was not evident in the convergence performance of QNS

for the free air calculations.

Figures 2.13a and b show the development of the corres-

ponding surface pressure distributions for the two cases discu-

ssed above. It is seen from Figure 2.13a that with $\alpha_1 = \alpha_2 = 0$,

the subsonic points attain their converged value (to plotting

accuracy) in as few as 10 iterations. The rest of the iterations

serve to adjust the location and strength of the shock. Even

so, the surface pressure distribution practically stabilises in

about 30 iterations. The convergence criterion for the present

calculation being rather stiff (about 0.1 percent on C_p), the
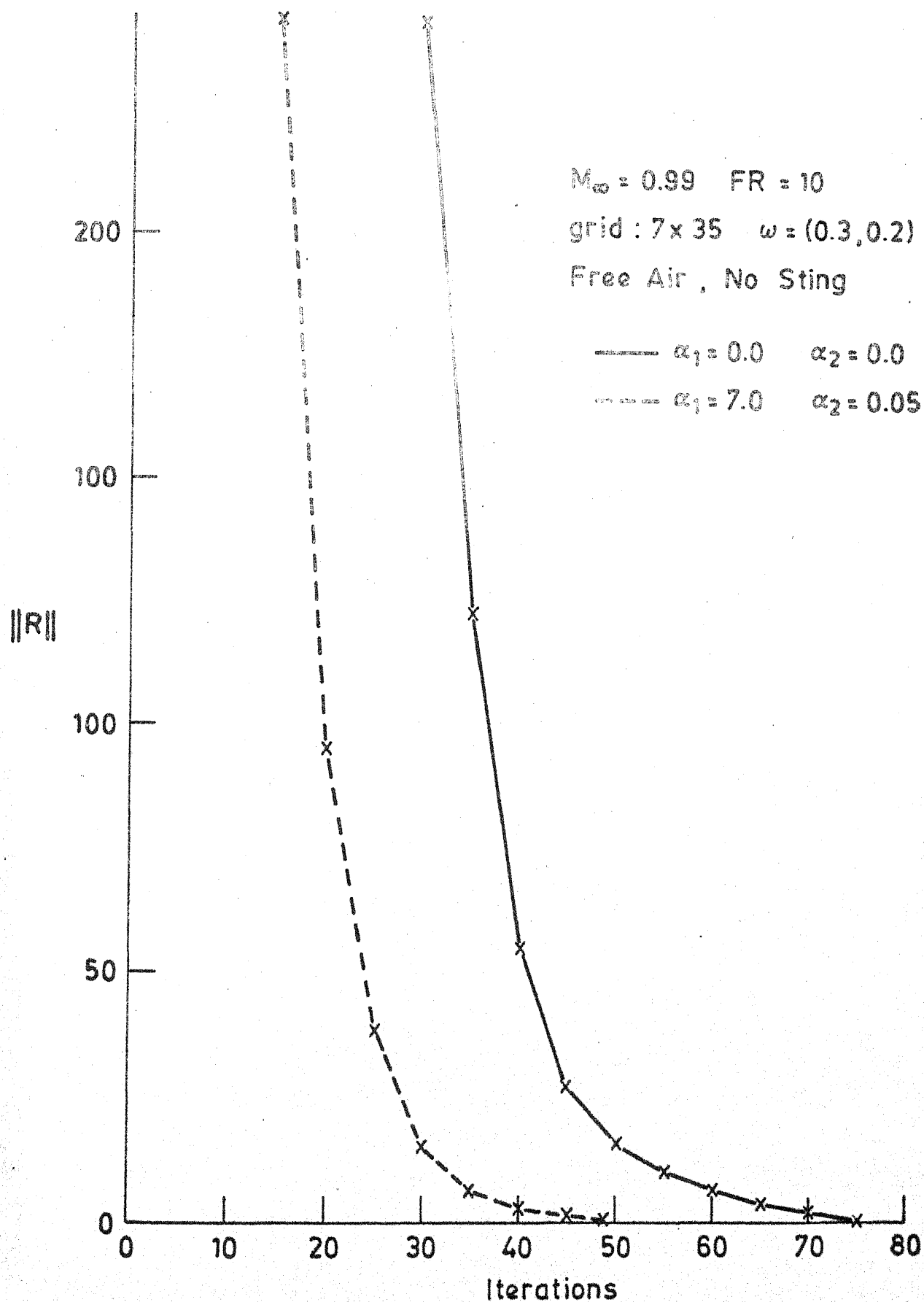
Fig. 2.12.   Effect of Iteration Parameters on Convergence History.
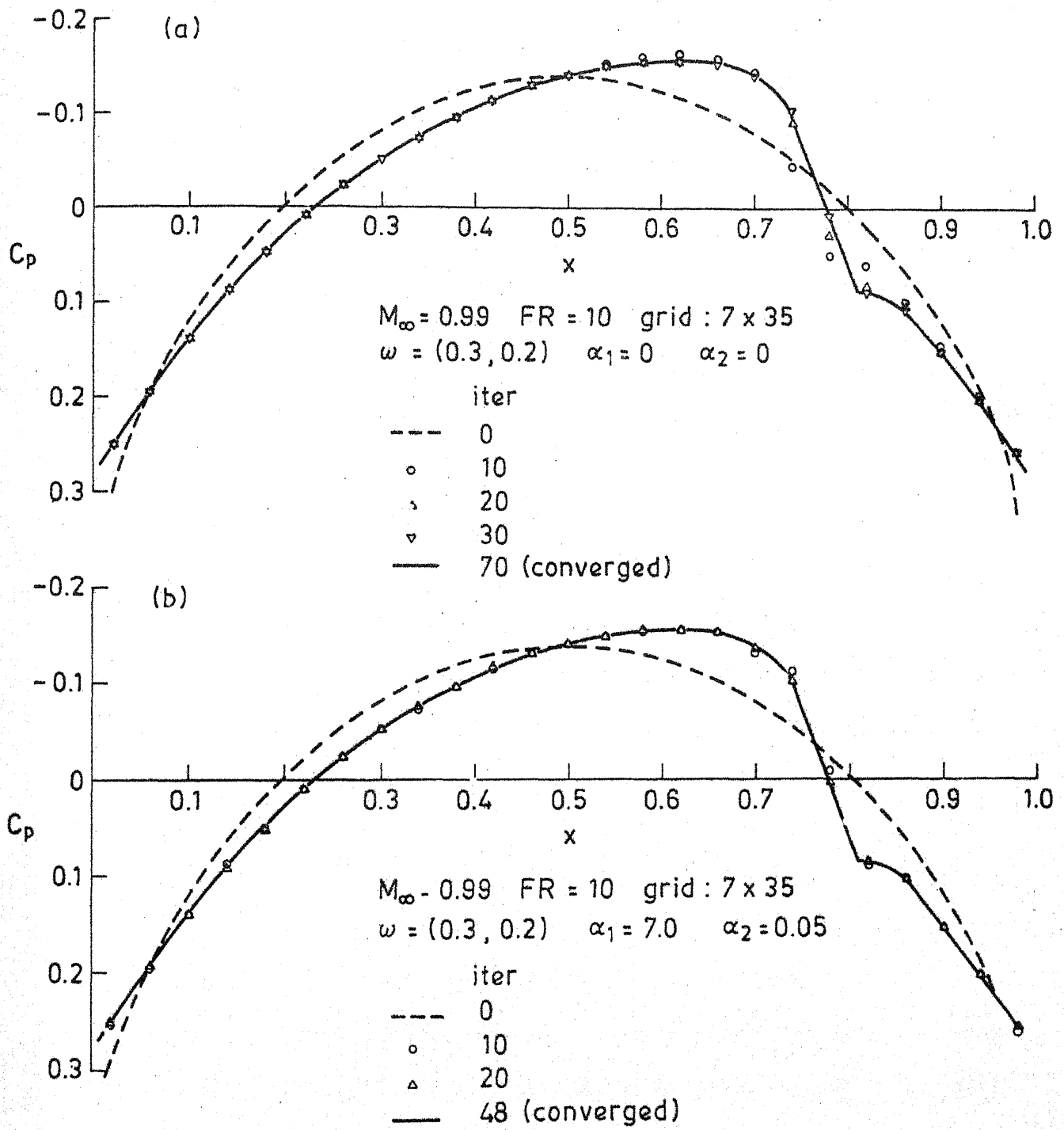
Fig. 2.13. Iterative Development : Surface Pressure Distribution on a Parabolic Body of Revolution in Free Air.

iterations were carried out for longer than was necessary to realise plotting accuracy which was about 1 percent.

Figure 2.13b shows the iterative development of the solution with $\alpha_1 = 7.0$ and $\alpha_2 = 0.05$. Introduction of the artificial time-dependent terms is seen to result in a practically converged solution being achieved in about 10 iterations. These figures also show that no large amplitude fluctuations develop in the solutions even for this fairly strong super-critical case, neither in the subsonic region nor in the super-sonic or in the shock regions. This indicates the possibility of attempting non-linear extrapolation of the solution to the converged solution after a sufficient number of iterations in the initial stages have been carried out.

Figure 2.14 shows similar results on the convergence characteristics of the DIS and QNS on u for the wall interference case. The free-stream Mach number is 0.99 and FR = 10. The porosity was set at p = 0.75. Four of the lines in Figure 2.14 show the error norm plotted against the number of iterations for DIS. The fifth line traces the convergence history of QNS. The damping factor ω for this case was (0.2, 0.3). It is to be noted here that the second value in the paranthesis is larger than the first. This was the only case for which a smaller value of ω for the supersonic points and larger value for the subsonic points failed to give convergence with or without the artificial time-dependent terms. Even values smaller than (0.2, 0.1) led to divergence of both DIS and QNS, whereas with ω = (0.2, 0.3), convergence could be achieved albeit only with the presence of the artificial time-dependent terms. Figure 2.14

$M_\infty = 0.99$ $FR = 10$ $p = 0.75$ with Sting
grid : 5 x 30 $\omega = (0.2, 0.3)$

	α_1	α_2	
— — — —	0.0	0.0	
————	2.0	0.0	DIS
·— ·— ·—	0.0	0.10	
————	1.5	0.10	
—— — ——	1.5	0.075	QRS

$\|R\|$
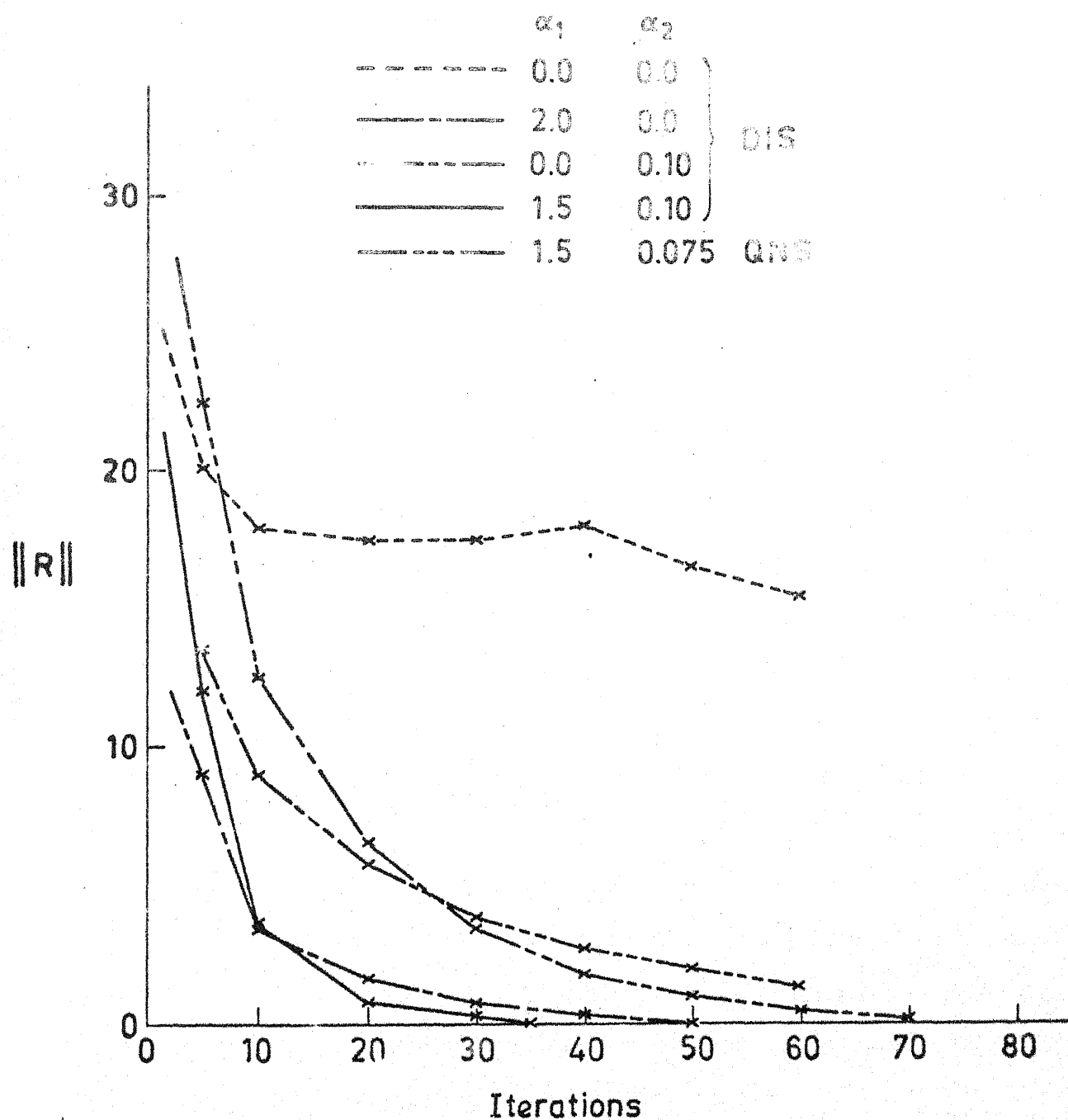
Iterations

Fig. 2.14.   Effect of Iteration Parameters on Convergence History for the Wall Interference Case.

shows that with $\alpha_1 = 1.5$, $\alpha_2 = 0.10$, convergence could be achieved in just 35 iterations. Comparison of this case with the one for $\alpha_1 = \alpha_2 = 0$ shows that convergence here is smooth and fast. The oscillations noticed in the behaviour of $\|R\|$ plot for $\alpha_1 = \alpha_2 = 0$ are damped out rapidly with the addition of the time-dependent terms. This is also confirmed by a study of the surface pressure plots shown in Figures 2.15a and b.

Figure 2.15a shows the development of the surface pressure distribution with $\alpha_1 = \alpha_2 = 0$. The appearance of a prominent non-uniformity in the surface pressure may be noticed just ahead of the shock at the end of about 20 iterations. At the end of 40 iterations, this non-uniformity has grown larger. The surface pressure thereafter tends to smoothen out as the iteration proceeds and the error norm correspondingly shows a decreasing trend. This case was found to converge extremely slowly as the iterations were carried on beyond 60. Figure 2.15b shows the development of the solution beginning with the initial solution. No non-uniformities appear in the solution as the iteration proceeds and convergence is apparently quite smooth. The solution at the end of 20 iterations is seen to be practically indistinguishable from the final converged solution which was achieved in 35 iterations.

The QNS also produced converged solutions for this case in about 90 iterations. The convergence history for this case is also displayed in Figure 2.14 and is seen to be inferior to that of the direct iteration scheme. This conclusion was also borne out by tests carried out for different values of the
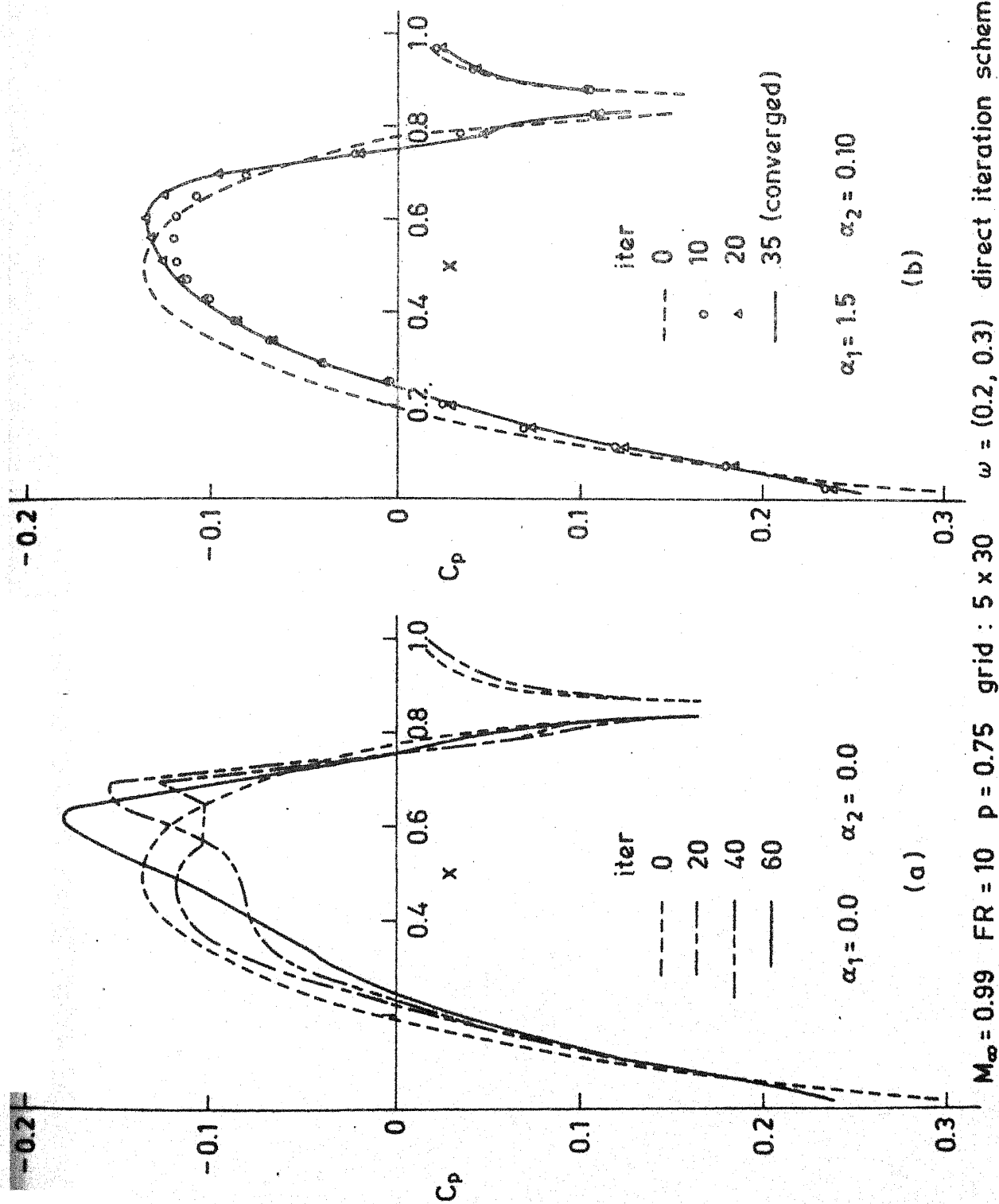
$M_\infty = 0.99$ $FR = 10$ $p = 0.75$ grid : 5×30 $\omega = (0.2, 0.3)$ direct iteration scheme

Fig. 2.15.   Iterative Development : Surface Pressure Distribution over a Parabolic Body of Revolution with Porous Wall Interference.
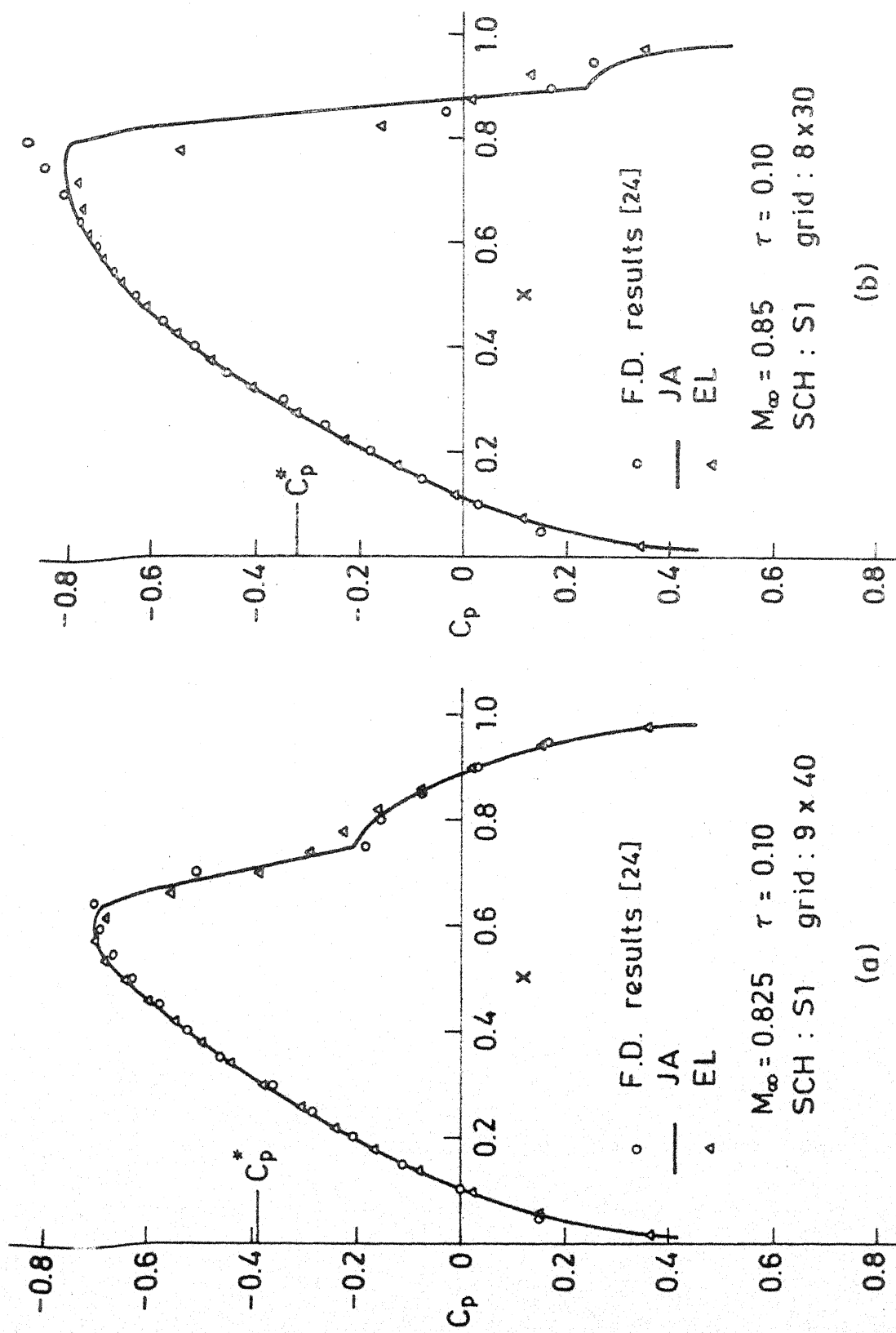
Fig. 2.16 Comparison of Surface Pressure Distribution with Finite Difference Results for a Parabolic Arc Airfoil.
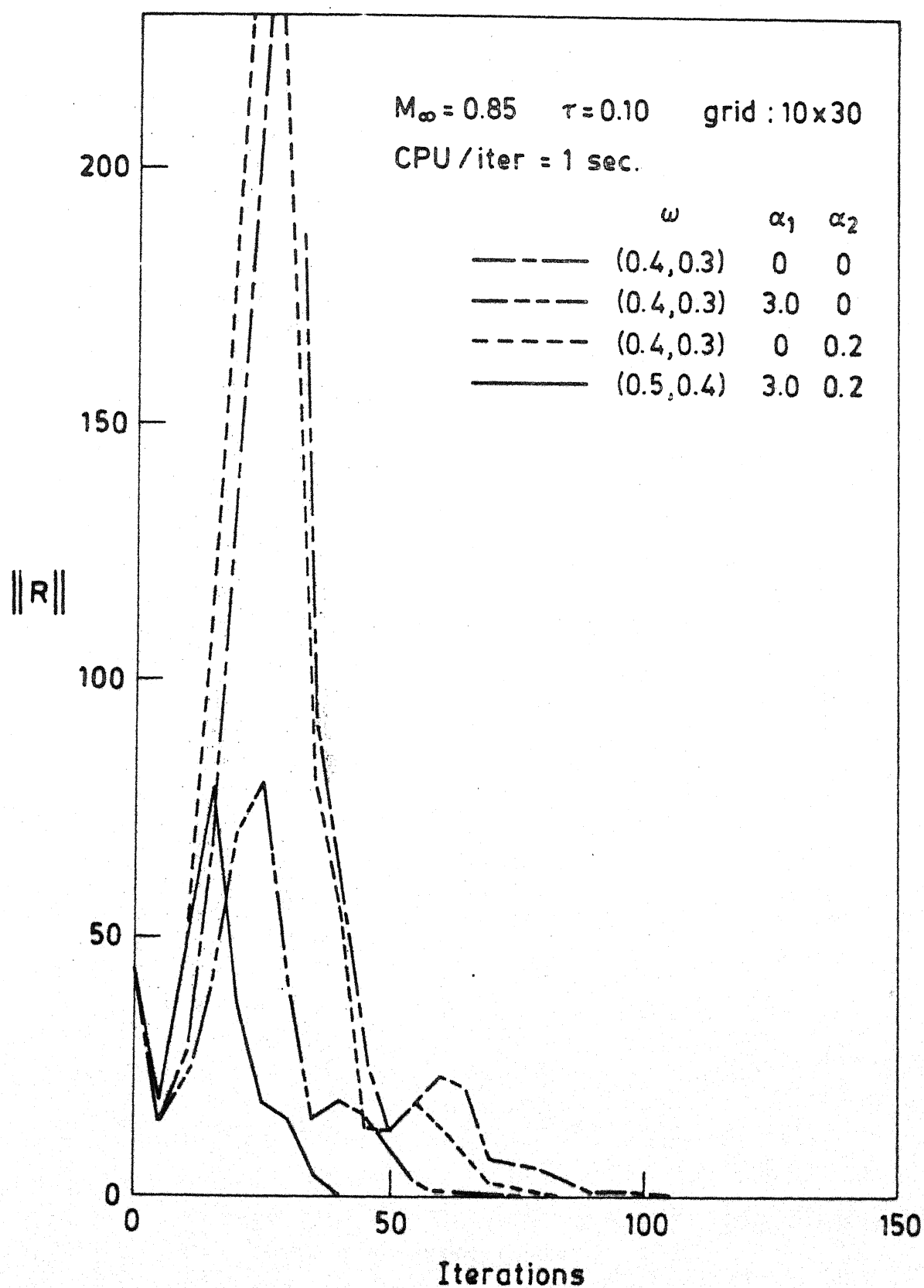
Fig. 2.17. Effect of Iteration Parameters on Convergence History for a Parabolic Airfoil.
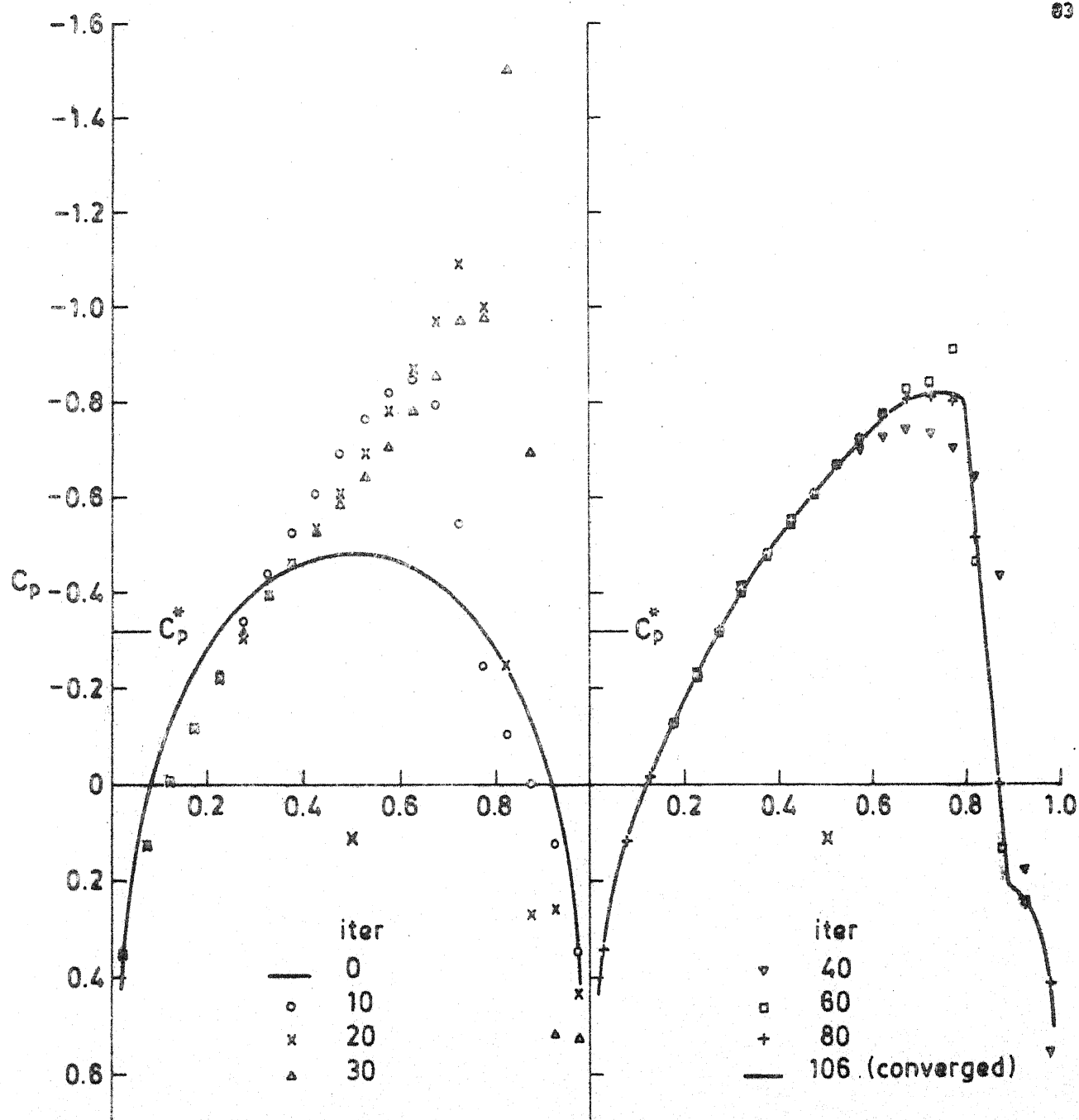
Fig. 2.18. Iterative Development : Surface Pressure Distribution on a Parabolic Airfoil. $M_\infty = 0.85$ $\tau = 0.10$ grid : 10×30 $\omega = (0.4, 0.3)$ $\alpha_1 = 0.0$ $\alpha_2 = 0.0$
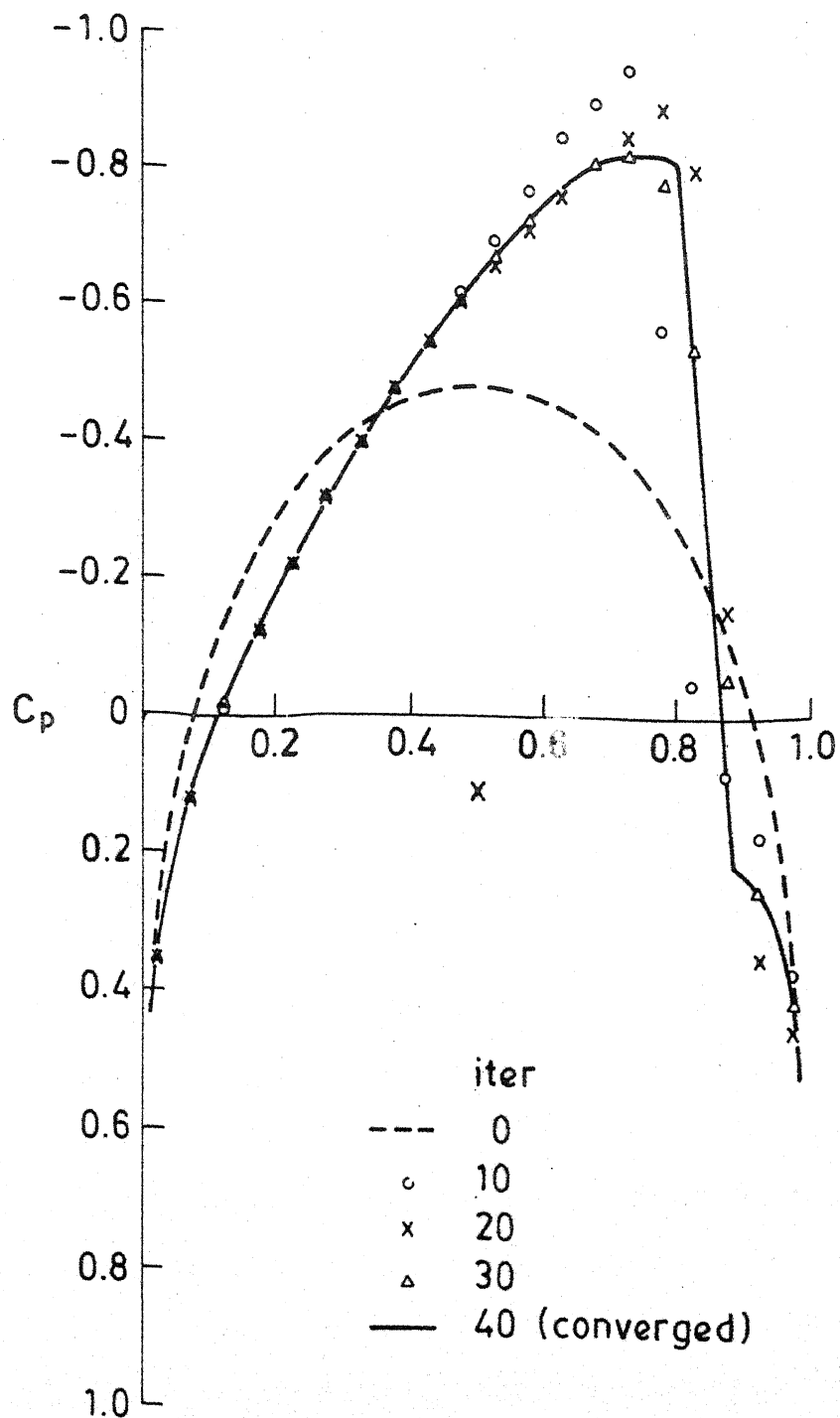
Fig. 2.19. Iterative Development : Surface Pressure
Distribution on a Parabolic Airfoil.
$M_\infty = 0.85$ $\tau = 0.10$ grid : 10 x 30
$\omega = (0.5, 0.4)$ $\alpha_1 = 3.1$ $\alpha_2 = 0.2$

flow parameters as well as various combinations of the values of ω, α_1 and α_2.

Admittedly this study of the artificial time-dependent damping terms merely demonstrates their utility in imparting stability to the iterative scheme. We have found yet no way of arriving at apriori estimates for the parameters ω, α_1 and α_2. The usefulness of these terms, in the absence of such estimates is restricted to achieving convergence for the stronger super-critical cases which might otherwise diverge. By a trial and error procedure, these cases may be made to converge by choosing appropriate values for ω, α_1 and α_2.

2.5.5. Some Results for Plane, Two-dimensional Symmetric Flows

A code analogous to the axisymmetric code was also developed for plane symmetric flow past airfoils. The results of Piers and Slooff [73] were in the main reproduced with this code. The results obtained using scheme S1 together with JA operator was found to give the best agreement with those of finite-difference calculations, as shown in Figure 2.16 for a 10 percent thick parabolic arc airfoil at $M_\infty = 0.825$ and 0.85. Numerical experiments were also carried out for this case by including artificial time-dependent terms. Reduction in the iteration counts comparable to those reported above for the axisymmetric case, were observed for plane flows also. These observations are summarized in Figures 2.17 to 2.19.

CHAPTER III

PLANE, TWO-DIMENSIONAL TRANSONIC FLOW PAST AIRFOILS — FULL-POTENTIAL SOLUTIONS

3.1. INTRODUCTION

In this chapter, we present a hybrid calculation procedure, combining the Integral Equation Method with elements of the finite-difference technique to compute full-potential solutions to two-dimensional transonic flows over airfoils with large embedded supersonic regions and weak shock discontinuities. In this approach, two simultaneous integral equations for the velocity components are formulated by considering the non-linear terms in the full-potential equation as a field source distribution, together with a distribution of sources and vortices on the mean line of the airfoil.

To be able to compute super-critical flows with shocks, an artificial viscosity term is added to the field source term in the supersonic regions so that expansion shocks are excluded from the computed solutions while the compression shocks are replaced by continuous regions with steep velocity gradients. Since the integral equations for the velocity components are coupled and non-linear, an iterative procedure is necessary for the numerical solution of these equations. The non-linearity consists in the presence of the field source term which is a function of the velocities and their derivatives. In order to calculate these derivatives, a body conforming mesh is generated through a sequence of co-ordinate transformations. The derivative

calculations are carried out on the transformed plane and the derivatives on the physical plane are obtained using the Jacobian of the transformation.

The iterative procedure starts by assuming the field source distribution to identically vanish and computes the incompressible solution to satisfy the tangency boundary condition on the body surface. The incompressible solution gives rise to a field source distribution which results in additional induced velocities, disturbing the boundary condition on the body surface. The internal singularity distribution is accordingly redetermined in order to satisfy the surface boundary condition iteratively. The non-linear problem thus reduces at every iteration to the solution of a Poisson problem. The procedure is repeated till convergence is achieved in the entire flow field.

The chapter concludes with a discussion of the computed solutions for sub-critical and super-critical flows past a few selected airfoils.

3.2. PROBLEM FORMULATION

3.2.1. The Inviscid Equations

Consider plane, two-dimensional flow past an arbitrary airfoil whose chord is aligned with the x-axis and the leading edge is located at the origin of a Cartesian co-ordinate system, Figure 3.1.

The equations governing compressible, inviscid and irrotational flow of a perfect gas can be written as

$\overline{\phi} \sim x \cos \alpha + y \sin \alpha$

$+\infty$

$+\infty$

$-\infty$

$+\infty$

$-\infty$

$-\infty$

$\dfrac{\delta \overline{\phi}}{\delta n} = 0$

$y = F_u(x)$

$y = F_l(x)$

$\underline{n}$

x

y

0

α

M_∞

1

$D:$

$\nabla \cdot (\rho \nabla \overline{\phi}) = 0$

$c^2 + \dfrac{\gamma - 1}{2} Q^2 = \dfrac{1}{M_\infty^2} + \dfrac{\gamma - 1}{2}$

$p = \rho^\gamma$
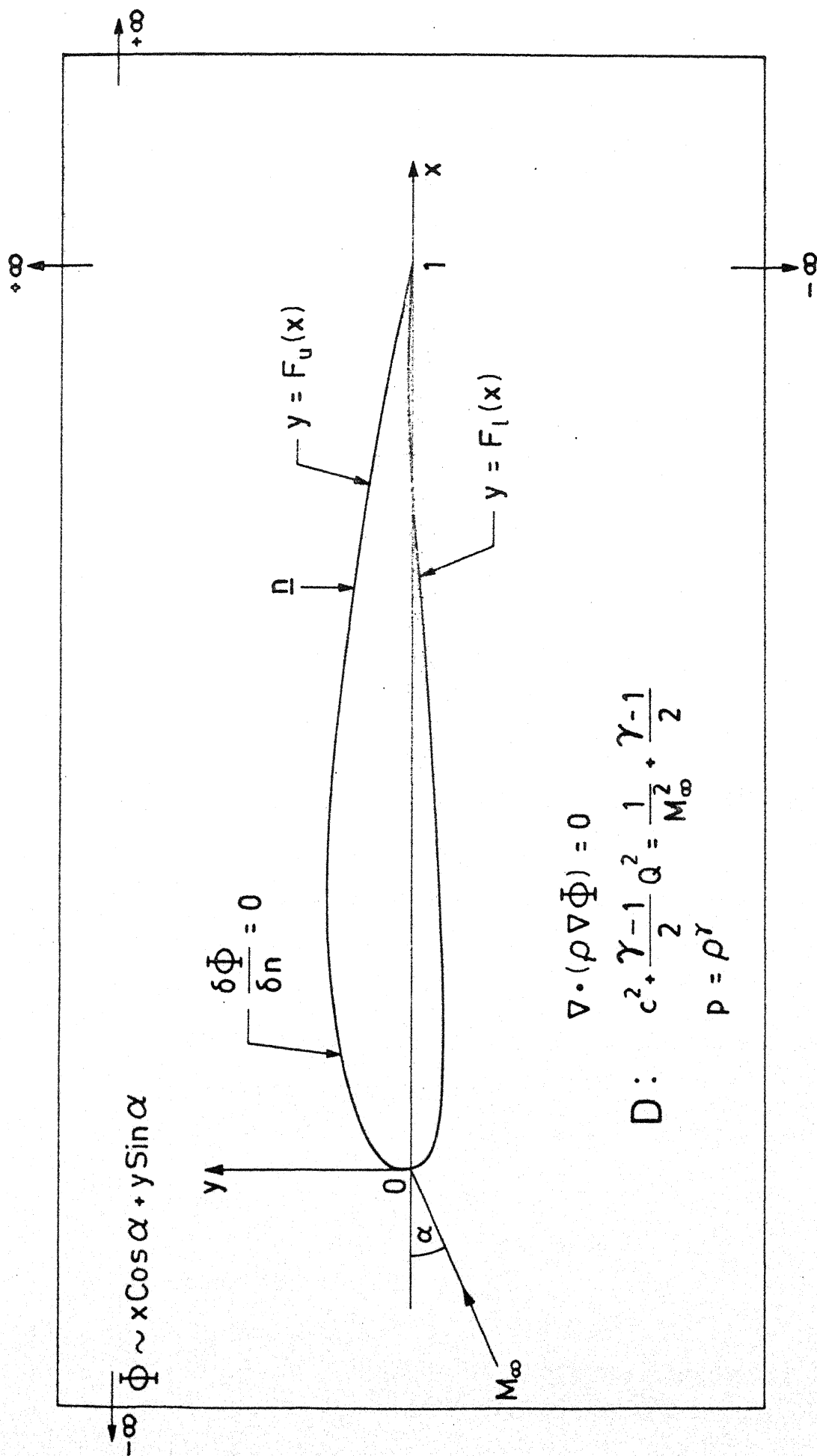
Fig. 3.1 Mathematical Model of Plane, Two-Dimensional, Compressible Flow Past an Airfoil.

Mass conservation : $\nabla \cdot (\rho \nabla \phi) = 0$ $\qquad$ (3.2.1)

Energy conservation: $c^2 + \dfrac{\gamma - 1}{2} Q^2 = \dfrac{1}{M_\infty^2} + \dfrac{\gamma - 1}{2}$ $\qquad$ (3.2.2)

where ϕ is the velocity potential, ρ the density, c the local speed of sound and Q the fluid speed.

For isentropic flows in which energy is conserved, it is not possible to conserve both mass and momentum across a shock. For an approximation in which mass is sought to be conserved, the momentum deficiency then provides an estimate of the wave drag.

To the conservation relations (3.2.1) and (3.2.2), the following isentropic relations may be added:

$$p = \rho^\gamma \qquad \text{and} \qquad c^2 M_\infty^2 = \rho^{\gamma - 1} \qquad (3.2.3)$$

In Eqs. (3.2.1) to (3.2.3), all velocities are made dimensionless by the free-stream velocity while pressure and density are made dimensionless by their respective free-stream values. The length scales are measured with the airfoil chord taken as unity.

Combining Eqs. (3.2.1) to (3.2.3), the so called gas dynamic equation is obtained

$$(c^2 - U^2) \phi_{xx} - 2UV \phi_{xy} + (c^2 - V^2) \phi_{yy} = 0 \qquad (3.2.4)$$

where U and V are the velocity components in the x and y directions and $Q^2 = U^2 + V^2$.

Equation (3.2.4) may also be written in the form

$$\phi_{xx} + \phi_{yy} = \frac{1}{c^2} \left[U \frac{\partial}{\partial x} \left(\frac{Q^2}{2} \right) + V \frac{\partial}{\partial y} \left(\frac{Q^2}{2} \right) \right] \equiv g \qquad (3.2.5)$$

where g, as a function of the velocities and their derivatives, represents the non-linear terms on the right acting as a field source distribution.

The total potential $\bar{\phi}$ may be split into a free-stream component and a perturbation component. Thus,

$$\bar{\phi}\ (x,\ y)\ =\ x\ \cos\alpha\ +\ y\ \sin\alpha\ +\ \emptyset(x,\ y) \qquad (3.2.6)$$

where $\emptyset$ is the perturbation potential and α is the angle of attack. Accordingly the velocities are given by

$$U\ =\ \cos\alpha\ +\ u, \qquad V\ =\ \sin\alpha\ +\ v \qquad (3.2.7)$$

where u and v denote the perturbation velocity components in the x and y directions respectively.

In terms of $\emptyset$, Eq. (3.2.5) simply becomes

$$\emptyset_{xx}\ +\ \emptyset_{yy}\ =\ g \qquad (3.2.8)$$

Let the airfoil shape be prescribed as

$$Y_{u,l}\ =\ F_{u,l}(x) \qquad 0\ \leqq\ x\ \leqq\ 1$$

where the subscripts u and l stand for the upper and lower surfaces respectively. For a solid surface, the body boundary condition is specified by setting the normal component of the flow velocity to zero on the body surface. In terms of the perturbation velocity components this can be expressed as

$$v\ -\ F'_{u,l}(x)u\ =\ \cos\alpha\ F'_{u,l}(x)\ -\ \sin\alpha$$

$$\text{on} \quad Y_{u,l}\ =\ F_{u,l}(x) \qquad (3.2.9)$$

where prime denotes differentiation with respect to x.

Since the perturbation velocity components must vanish at infinity we have

$$u, v \rightarrow 0 \quad \text{as} \quad (x^2 + y^2)^{1/2} \rightarrow \infty \qquad (3.2.10)$$

In addition to Eqs. (3.2.9) and (3.2.10), the Kutta condition of smooth flow at the trailing edge must be satisfied.

The local pressure co-efficient is calculated from

$$C_p = \frac{2}{\gamma M_\infty^2} [\{1 - \frac{\gamma - 1}{2} M_\infty^2 (Q^2 - 1)\}^{\frac{\gamma}{\gamma - 1}} - 1] \qquad (3.2.11)$$

3.2.2. Super-critical Flows, Artificial Viscosity

For super-critical flows, as in the TSPE formulations, it is necessary to add an artificial viscosity to the governing equation in order to exclude expansion shocks and capture only the compression shocks. To arrive at an appropriate artificial viscosity model for the integral equation method, it will be instructive to study the way in which artificial viscosity is introduced in finite-difference calculation schemes [25].

The basic idea behind Murman's [5] mixed differencing scheme is to introduce directional bias as well as to incorporate the correct domain of dependence in the difference scheme at supersonic points. In the small disturbance formulation, where the flow direction is assumed to be more or less aligned with the x-co-ordinate axis, this is accomplished by simple upwind differencing of the x-derivatives at the supersonic points (see Section 2.2.4). For the full-potential formulation

however, the upwind direction is not known in advance. Nevertheless, for flows with supersonic zones of moderate size, satisfactory results have been obtained by the use of a simple upwind differencing scheme in which central difference formulae are used at subsonic points for all derivatives and upwind formulas for ϕ_{xx} and ϕ_{xy} at the supersonic points. The upwind differencing of Eq. (3.2.4) at the supersonic points then introduces an effective artificial viscosity

$$\Delta x \left[(U^2 - c^2) \, U_{xx} + UVV_{xx} \right]$$

When the flow is not perfectly aligned with the x-axis, there exist supersonic points at which $U^2 < c^2 < U^2 + V^2$. One characteristic lies ahead of the y-co-ordinate line at such a point, so that the difference scheme does not have the correct region of dependence, see Figure 3.2a. Correspondingly, the artificial viscosity $\Delta x(U^2 - c^2)U_{xx}$, introduced by the use of the upwind formula at such a point becomes negative. Despite this fact, schemes of this type have proved satisfactory for flows with moderate supersonic zones.

The treatment of flows with large supersonic zones in a curvilinear co-ordinate system, suited to the geometry of the problem, requires the use of a more elaborate difference scheme in which the direction of upwind differencing is independent of the co-ordinate system and is rotated to conform to the local flow direction. Setting up a local Cartesian co-ordinate system and denoting the local streamwise and normal directions by the s and n co-ordinates, the quasi-linear equation (3.2.4) can be written as
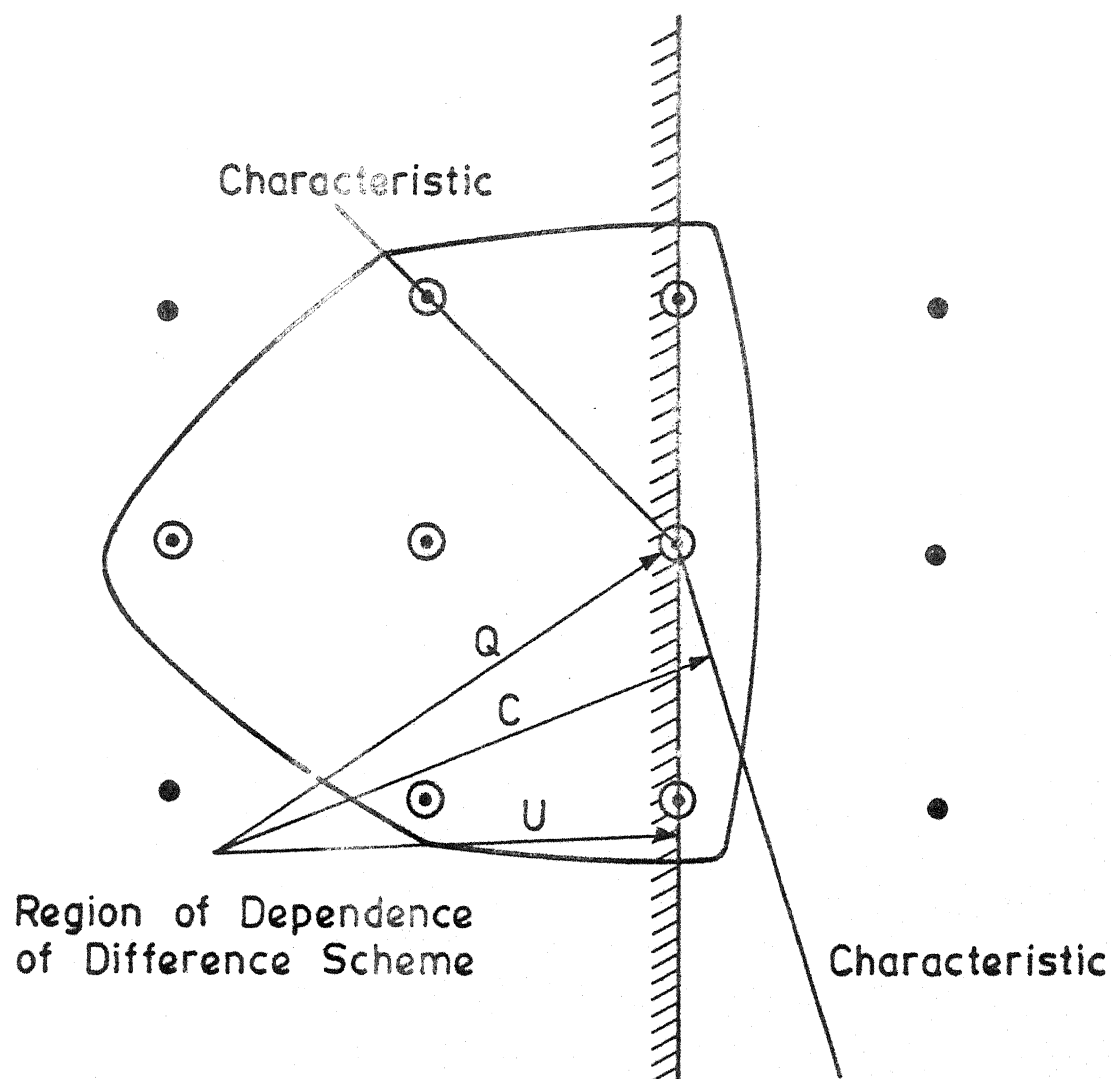
Fig. 3.2a.   Simple Upwind Difference Scheme; [25]

$$(c^2 - Q^2) \, \phi_{ss} + c^2 \, \phi_{nn} = 0 \qquad\qquad (3.2.12)$$

Since U/Q and V/Q are the local direction cosines, ϕ_{ss} and ϕ_{nn} may be expressed in terms of the original co-ordinates as

$$\phi_{ss} = \frac{1}{Q^2} (U^2 \, \phi_{xx} + 2UV \, \phi_{xy} + V^2 \, \phi_{yy}) \qquad (3.2.13)$$

$$\phi_{nn} = \frac{1}{Q^2} (V^2 \, \phi_{xx} - 2UV \, \phi_{xy} + U^2 \, \phi_{yy}) \qquad (3.2.14)$$

Then at subsonic points, central difference formulas are used for both ϕ_{ss} and ϕ_{nn}. At supersonic points, ϕ_{nn} is calculated using central differences, but upwind formulas are used for evaluating all the derivatives occurring on the right of Eq. (3.2.13). Such a scheme consequently has the correct region of dependence at the supersonic points as shown in Figure 3.2b. Upwind differencing in this case introduces an effective artificial viscosity

$$(1 - \frac{c^2}{Q^2}) \, [\, \Delta x(U^2 U_{xx} + UVV_{xx}) + \Delta y(UVU_{yy} + V^2 V_{yy})] \qquad (3.2.15)$$

which is symmetric in x and y.

In the integral equation formulations, the artificial viscosity term is to be added explicitly to the field source term. This objective is realised by modelling the added artificial viscosity term to correspond to that given by Eq. (3.2.15). Going back to Eq. (3.2.12), we see that if at the supersonic points ϕ_{ss} is represented by upwind difference formulas, this is equivalent to the addition of an artificial viscosity
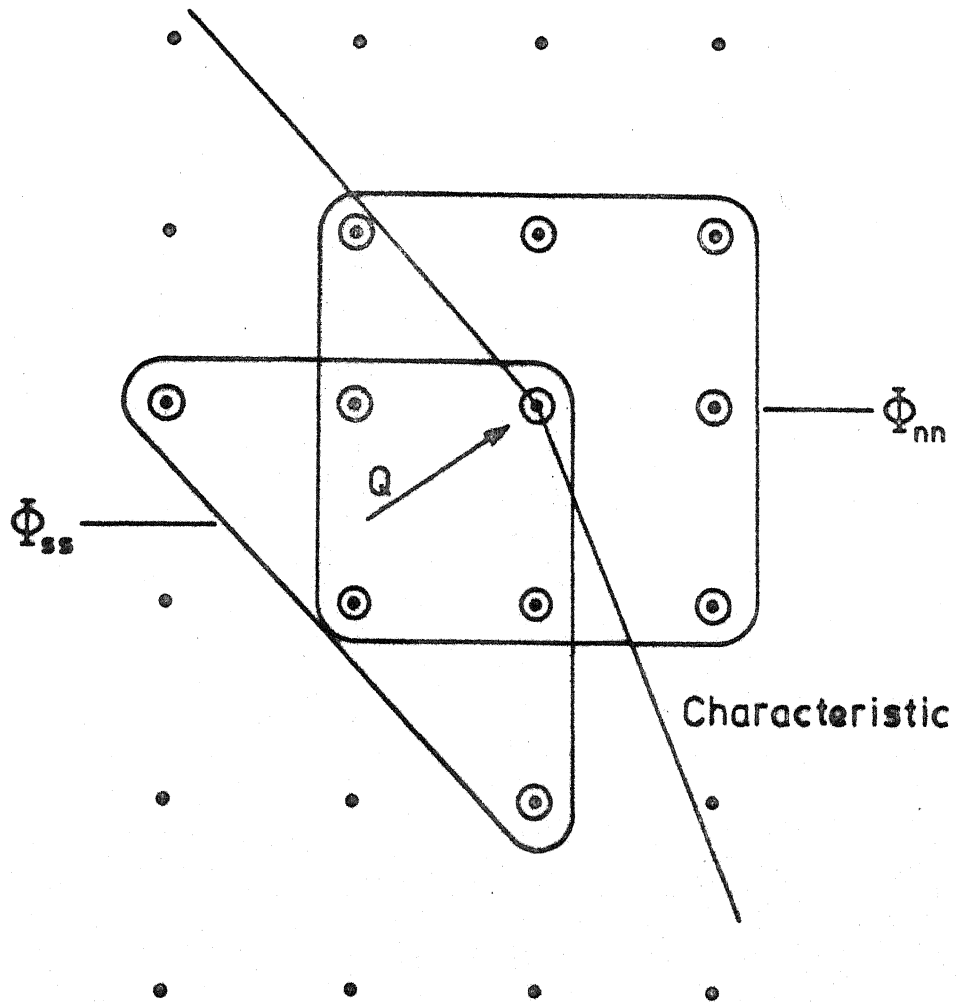
Fig. 3.2b. Rotated Difference Scheme, [25]

$$T = - \Delta s \, \mu \, Q^2 \, \phi_{sss} \qquad (3.2.16)$$

where μ is a switching function defined as

$$\mu = \max \left[0, \, 1 - \frac{c^2}{Q^2} \right] \qquad (3.2.17)$$

Since the mass continuity equation is equivalent to multiplying Eq. (3.2.4) by ρ/c^2, the conservative form for the artificial viscosity term may be approximated as

$$T = - \frac{\partial}{\partial s} \left[\Delta s \, \mu \, \frac{\rho}{c^2} \, Q \, \frac{\partial}{\partial s} \left(\frac{Q^2}{2} \right) \right] \qquad (3.2.18)$$

It may be verified that the highest derivatives in the leading term of Eq. (3.2.18) are the same as those in Eq. (3.2.15), see Appendix D.

Equation (3.2.18) is written in terms of the streamwise derivatives and hence can be applied conveniently on the surface grid points. For the interior points however, it is necessary to write the artificial viscosity in terms of the x and y derivatives as (see Appendix D).

$$T = \frac{\partial G}{\partial x} + \frac{\partial H}{\partial y} \qquad (3.2.19)$$

where the artificial viscosity functions G and H are defined as

$$G = - \Delta x \, \frac{\rho}{c^2} \, \mu \, U \, \frac{\partial}{\partial x} \left(\frac{Q^2}{2} \right) \qquad (3.2.20)$$

$$H = - \Delta y \, \frac{\rho}{c^2} \, \mu \, V \, \frac{\partial}{\partial y} \left(\frac{Q^2}{2} \right) \qquad (3.2.21)$$

With the addition of the artificial viscosity term to the right of Eq. (3.2.5), the modified field source term may be written as

$$\bar{g} = g(x, y) + T \qquad\qquad (3.2.22)$$

where T is given by either (3.2.18) or (3.2.19).

In Eq. (3.2.22) it is noted that while the inviscid term g is not in conservation form, the artificial viscosity term is in conservation form. Hence a scheme based on Eq. (3.2.22) may be said to correspond to the quasi-conservative scheme of Bauer et al [99].

3.2.3. Artificial Compressibility

The fact that the inviscid field source term may also be expressed as $g = - Q/\rho \frac{\partial \rho}{\partial s}$, suggests that the idea of artificial compressibility may be just as easily implemented in the present scheme as artificial viscosity. The relative merits of the two approaches in a numerical procedure are not clear yet although Harten[100] has claimed that the artificial compressibility method is superior to artificial viscosity, in particular for capturing contact discontinuities.

A simple approach to introducing artificial compressibility in the inviscid equations is to modify the continuity equation as

$$\frac{\partial}{\partial x} (\bar{\rho} \, \hat{z}_x) + \frac{\partial}{\partial y} (\bar{\rho} \, \hat{z}_y) = 0 \qquad\qquad (3.2.23)$$

where the modified density is given by

$$\bar{\rho} = \rho - \mu \, \Delta s \, \rho_s \qquad\qquad (3.2.24)$$

Equation (3.2.24) may be arrived at as follows.

The artificial viscosity function given by Eq. (3.2.20) is

$$G = -\Delta x \frac{\rho}{c^2} \mu U \frac{\partial}{\partial x} \left(\frac{Q^2}{2}\right)$$

$$= \Delta x \mu U \frac{\partial \rho}{\partial x} \quad \text{since} \quad \frac{\partial \rho}{\partial x} = -\frac{\rho}{c^2} \frac{\partial}{\partial x} \left(\frac{Q^2}{2}\right)$$

Similarly

$$H = \Delta y \mu V \frac{\partial \rho}{\partial Y}$$

The form of these two functions immediately suggests writing the artificial compressibility term in the form given by Eq. (3.2.24).

The modified field source term $\bar{g}$ for the artificial compressibility method is then

$$\bar{g} = -\frac{Q}{\rho} \frac{\partial \bar{\rho}}{\partial s} \tag{3.2.25}$$

with $\bar{\rho}$ defined by Eq. (3.2.24).

Other artificial compressibility terms may be found discussed in Ref. [101].

3.3. THE INTEGRAL EQUATION

For the purpose of formulating the integral equation for the perturbation potential, Eq. (3.2.5) may be regarded as a Poisson equation with a field source term $\bar{g}$ given by Eqs. (3.2.22) or (3.2.25). The solution for $\emptyset$ may then be expressed as the sum of a general solution satisfying the homogeneous part of the equation and a particular solution.

Since the general solution satisfies the Laplace equation, it may be built up by a suitable distribution of singularities. For lifting flows over two-dimensional airfoils, a

combination of sources and vortices are distributed on the mean line of the airfoil, as shown in Figure 3.3. The non-linear term $\bar{g}$ can be thought of as a field source distribution. Hence the potential induced at a field point (x, y) is the sum of the potentials induced by the internal singularities and the field sources. Assuming the source density to be $\sigma(x)$, vortex density $\gamma(x)$, we may write

$$
\begin{aligned}
\emptyset(x, y) = & \int_C K_S(x - x', y - y_m(x'))\, \sigma(x', y_m(x'))\, dl \\
& + \int_C K_V(x - x', y - y_m(x'))\, \gamma(x', y_m(x'))\, dl \\
& + \iint_D K_S(x - x', y - y')\, \bar{g}(x', y')\, dx'\, dy'
\end{aligned}
$$

$$(3.3.1)$$

where the kernels are given by

$$
K_S(x - x', y - y') = \frac{1}{2\pi} \ln \left[(x - x')^2 + (y - y')^2 \right]^{1/2}
$$

$$(3.3.2)$$

$$
K_V(x - x', y - y') = \frac{1}{2\pi} \tan^{-1} \frac{y - y'}{x - x'}
$$

$$(3.3.3)$$

In Eq. (3.3.1), D is the entire plane excluding the area enclosed by the airfoil contour. The line integrals are over the curve C which denotes the mean line $y_m(x)$ on which the singularities are distributed. A small gap δ is maintained between the leading edge and the tip of the mean line so as to render the velocities finite at the leading edge. In Eq. (3.3.1) hence, the first two terms represent the potential induced by the source and vorticity distributions, while the last term represents the effects of compressibility.
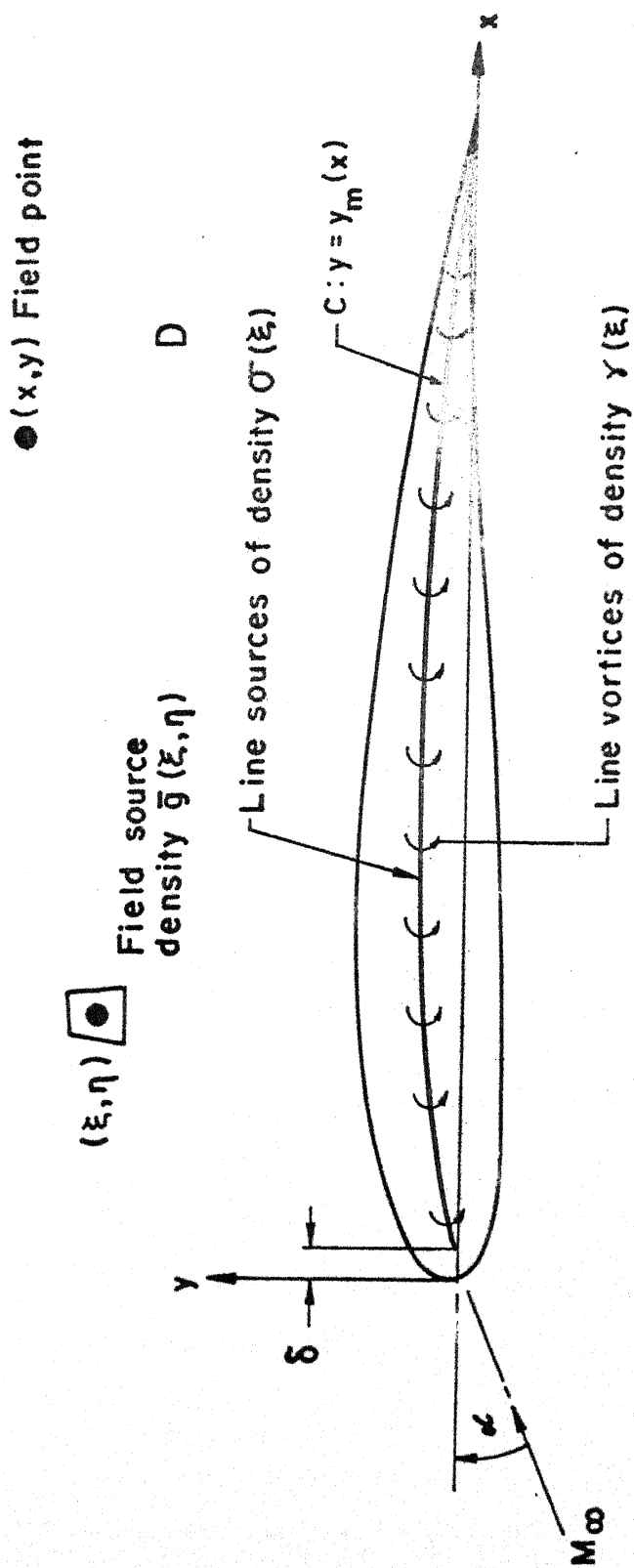
Fig. 3.3   Singularity Model for Airfoil in Compressible Flow.

The perturbation potential given by Eq. (3.3.1) satisfies the differential equation (3.2.5) in the region D with the field source term replaced by $\bar{g}$, as can be verified by direct differentiation. Hence $\emptyset$ is a solution of Eq. (3.2.5) in D.

The integral equation for the perturbation velocity components may be obtained by differentiating the expression for $\emptyset$. For brevity we adopt the following functional notation and write

$$\emptyset(x, y) = \mathcal{H}(K_S, K_V)$$

$$u(x, y) = \mathcal{H}(K_{Sx}, K_{Vx}) \tag{3.3.4}$$

$$v(x, y) = \mathcal{H}(K_{Sy}, K_{Vy}) \tag{3.3.5}$$

where the subscripts x and y denote the respective partial derivatives of the kernels.

Equations (3.3.4) and (3.3.5) constitute a system of two integral equations for the perturbation velocity components u and v, which must be solved by satisfying the tangency boundary condition Eq. (3.2.9) on the surface of the airfoil. The far-field boundary condition is implicitly satisfied by the integral equations as can be verified for the first two terms by inspection and, for the field integral by making appropriate assumptions about the asymptotic behaviour of $\bar{g}$.

The solution of the Eqs. (3.3.4) and (3.3.5) is to be obtained by numerical methods.

3.4. NUMERICAL SOLUTION

A glance at Eqs. (3.3.1), (3.3.4) and (3.3.5) suggests that it may be advantageous to solve Eq. (3.3.1) for $\emptyset$ because it involves the computation of fewer number of influence co-efficients. However in a hybrid calculation procedure of the kind attempted here, accurate evaluation of velocities from the perturbation potential using finite-difference derivatives would demand a rather close spacing of the grid lines adjacent to the airfoil as in finite-difference methods. On the other hand, dealing with Eqs. (3.3.4) and (3.3.5) directly, involving the computation of the influence co-efficients for both the velocity components, permits accurate evaluation of the same, especially at the surface points. This is particularly important from the point of view of satisfying the boundary condition on the body surface, if satisfactory accuracy is to be achieved on a coarse grid.

It is possible to discretize the coupled system of Eqs. (3.3.4) and (3.3.5) on the physical plane, using rectangular panels with sides parallel to the co-ordinate axes. The drawback with this procedure is that the special care necessary to cluster mesh cells in the leading edge region, makes the grid construction airfoil dependent and complicates input data specification. Moreover such a mesh leads to uneconomical storage and inefficient coding, especially for lifting flows. A curvilinear grid generated through well defined analytical and numerical transformations does away with these difficulties.

3.4.1. Grid Generation

The following sequence of transformations used by Mercer et.al. [102] is defined to construct the $\nu - \theta$ plane from the physical x - y plane, see Figure 3.4.

$$\bar{x} = x - r_t \qquad \bar{y} = y \qquad (3.4.1a)$$

$$w = \cosh^{-1}(1 - 2\exp(z)) \qquad (3.4.1b)$$

$$\nu = s \qquad \theta = t/t_{upper} \qquad (3.4.1c)$$

$$w = s + it \qquad \text{and} \qquad z = \bar{x} + i\bar{y}$$

The principal transformation of course is Eq. (3.4.1b). It unwraps the airfoil from the trailing edge and stretches it on either side of the t-axis, on the s - t plane. r_t is the leading edge radius if the airfoil is round nosed or a small positive value if the airfoil is sharp nosed. Eq. (3.4.1a) simply shifts the origin from the leading edge of the airfoil to the centre of curvature of the leading edge. This is necessary because the second transformation is singular at the origin which must therefore be excluded from the computational domain.

The last transformation (3.4.1c) yields a rectangular domain on which the upper and lower surfaces of the airfoil are stretched into $\theta = 1$ lines on either side of the θ-axis. The upstream infinity maps on to the origin of the $\nu - \theta$ plane. The downstream infinities of the upper and lower surface wakes along the positive x-axis on the physical plane, correspond

$$\bar{x} = x - r_t$$
$$\bar{y} = y$$

(a)

Centre of curvature

$$w = s + it = \text{Cosh}^{-1}\left(1 - 2\,\exp\,(z)\right)$$
$$z = \bar{x} + i\bar{y}$$

upper wake surface   lower wake surface

T.E.upper   T.E. lower

π

LE

Airfoil
upper surface

Airfoil
lower surface

t

O   s

(b)   Upstream
infinity

$$\nu = s\ ,\theta = t/t_{upper}$$

θ

O ν

(c)
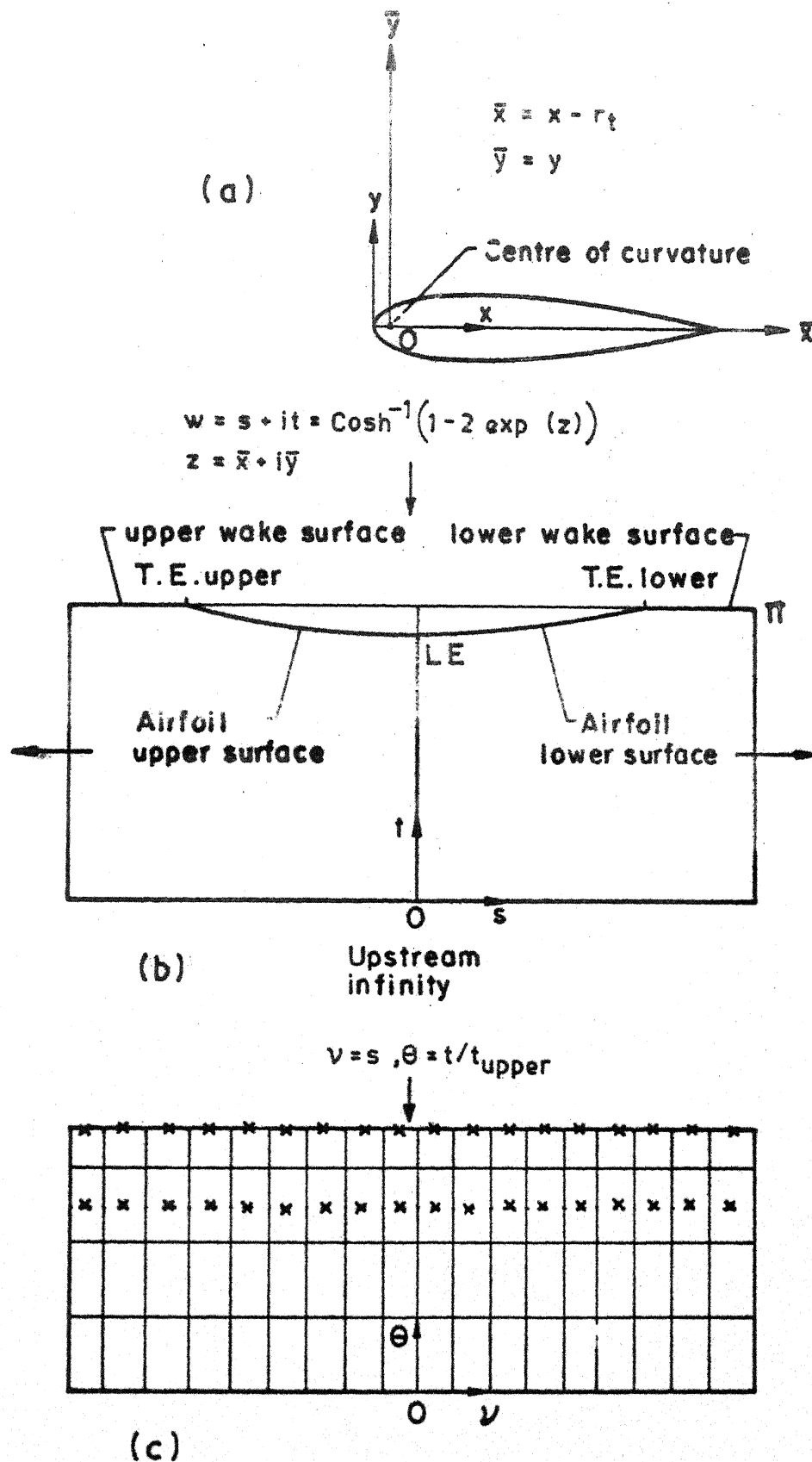
Fig. 3.4.   Grid generation through co-ordinate transformations
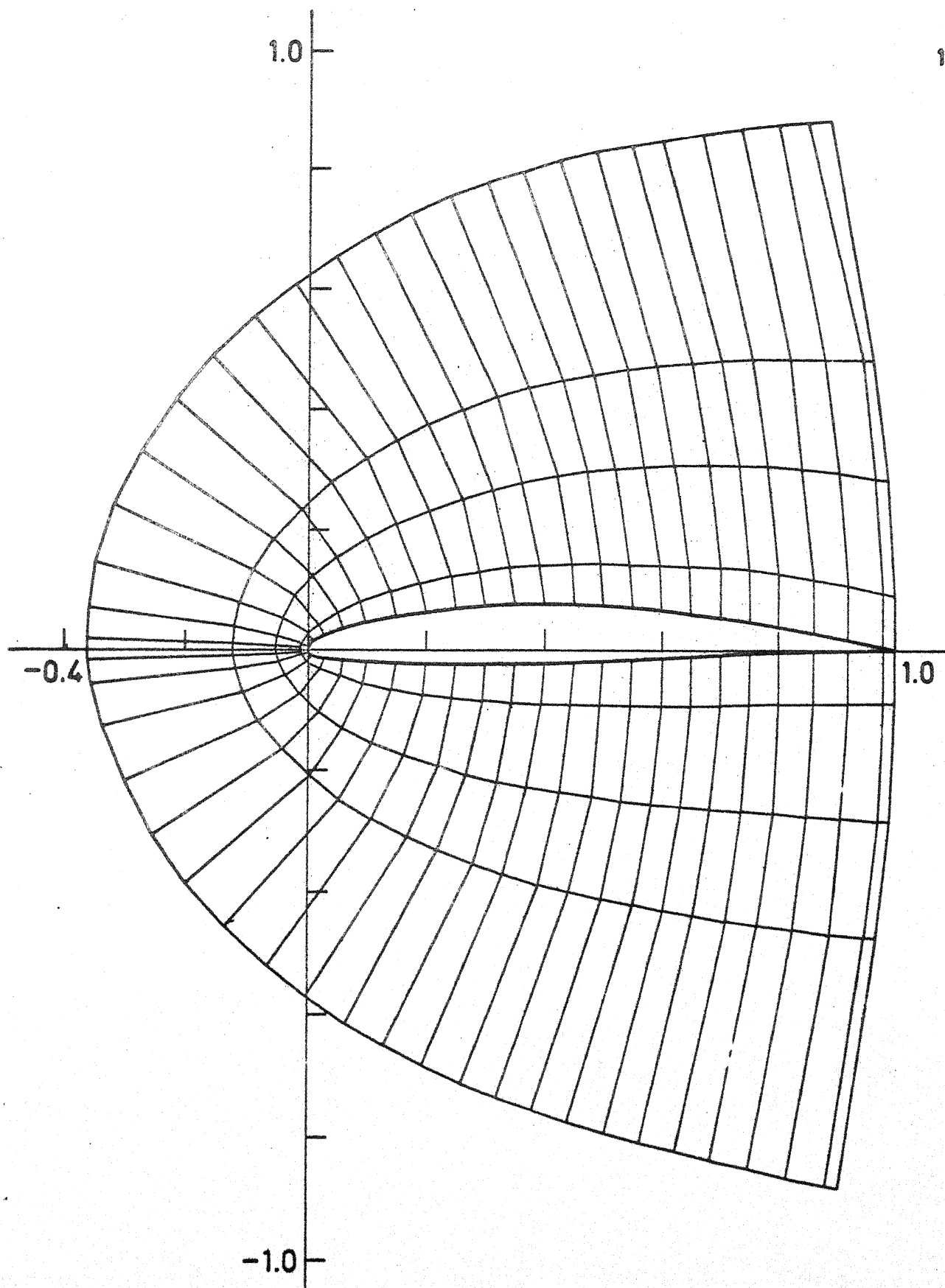
Fig. 3.5. Grid on the Physical Plane for a NACA 64A410 Airfoil.
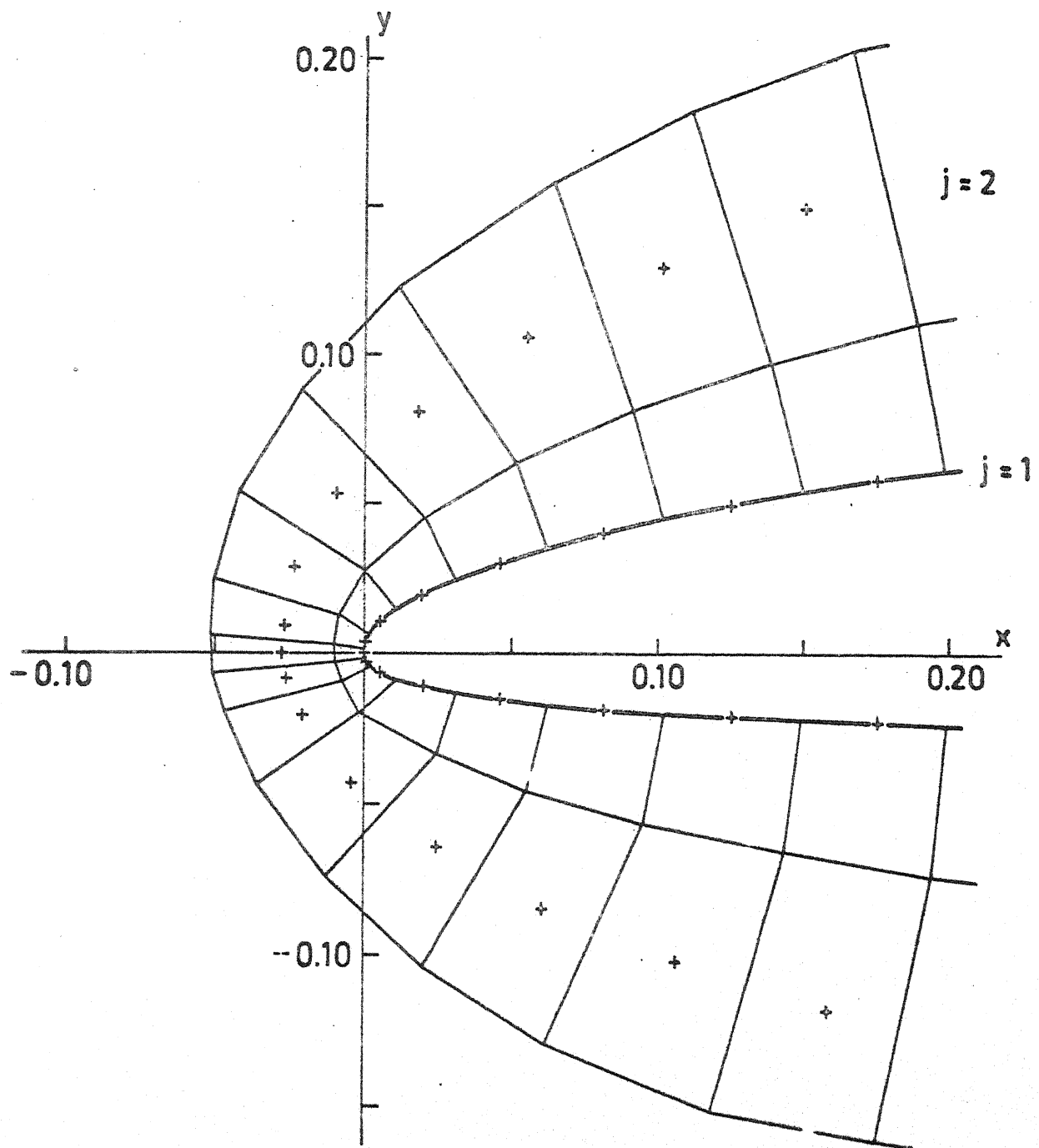
Fig. 3.5 (contd). Grid Details Near the Leading Edge.

respectively to the negative and positive infinities of the
ν-axis in Figure 3-4.

The rectangular domain on the $\nu - \theta$ plane is divided
into smaller rectangular cells. The rectangular grid is now
mapped back to the physical plane yielding constant ν and θ
lines, which constitute the curvilinear grid on the latter
plane; Figure 3.5. All derivative calculations are carried
out on the $\nu - \theta$ plane and the derivatives on the physical
plane are obtained using the Jacobian of the transformation,
according to the formula

$$J \left[\frac{\partial}{\partial \nu} \quad \frac{\partial}{\partial \theta} \right]^T = \left[\frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \right]^T \tag{3.4.2}$$

where the Jacobian J is given by

$$J = \begin{bmatrix} \dfrac{\partial \nu}{\partial x} & \dfrac{\partial \theta}{\partial x} \\[2ex] \dfrac{\partial \nu}{\partial y} & \dfrac{\partial \theta}{\partial y} \end{bmatrix} \tag{3.4.3}$$

and T denotes the transpose.

The computation of J is carried out as follows.

$$J = \begin{bmatrix} \dfrac{\partial s}{\partial x} & \dfrac{\partial t}{\partial x} \\[2ex] \dfrac{\partial s}{\partial y} & \dfrac{\partial t}{\partial y} \end{bmatrix} \begin{bmatrix} 1 & - tt'_u(s)/t_u^2(s) \\[2ex] 0 & 1/t_u(s) \end{bmatrix} \tag{3.4.4}$$

where $t_u(s)$ denotes the profile surface on the $s - t$ plane and
prime denotes differentiation with respect to s. Since the
transformation (3.4.1b) is analytic on the $\bar{x} - \bar{y}$ plane except
at the origin, its real and imaginary parts satisfy the Cauchy-
Riemann equations. Hence

$$\frac{\partial t}{\partial x} = -\frac{\partial s}{\partial y} \quad \text{and} \quad \frac{\partial s}{\partial x} = \frac{\partial t}{\partial y} \qquad (3.4.5)$$

Substituting in Eq. (3.4.4) and carrying out the matrix multiplication we obtain

$$J = \begin{bmatrix} \dfrac{\partial s}{\partial x} & -\dfrac{\Theta t'_u(s)}{t_u(s)}\dfrac{\partial s}{\partial x} + \dfrac{1}{t_u(s)}\dfrac{\partial t}{\partial x} \\[3mm] -\dfrac{\partial t}{\partial x} & \dfrac{\Theta t'_u(s)}{t_u(s)}\dfrac{\partial t}{\partial x} + \dfrac{1}{t_u(s)}\dfrac{\partial s}{\partial x} \end{bmatrix} \qquad (3.4.6)$$

The derivatives $\partial s/\partial x$, $\partial t/\partial x$ and $t'_u(s)$ are all evaluated using second order accurate central difference formulae. The Jacobian J is thus evaluated for each grid point and stored in memory.

3.4.2. Discretization

Equations (3.3.4) and (3.3.5) will now be discretized by replacing the integrals by sums and derivatives by differences.

Field Sources

In each cell of Figure 3.5, formed by the intersection of the constant ν and Θ lines on the physical plane, we consider all the flow variables to be uniform. Each cell is approximated by a quadrilateral and its influence co-efficients can be evaluated at every field point analytically, following a method outlined by Hess [103]. The field velocity contributions may be obtained from the sum formulae

$$u_F(x, y) = \sum_k \sum_l e^{kl}(x, y)\, \bar{g}_{kl} \qquad (3.4.7)$$

$$v_F(x, y) = \sum_k \sum_l f^{kl}(x,y) \, \bar{q}_{kl} \qquad (3.4.8)$$

where e and f denote the influence co-efficients for the field sources. Summation indices k and l respectively denote grid points running along ν and θ directions. The expressions for the influence co-efficients e and f are derived in Appendix E.

Internal Singularities

To carry out the discretization of the mean line, the airfoil chord is first divided into three main segments as shown in Figure 3.6. Each segment is now further subdivided into smaller elements, according to the distribution given in the figure, by specifying the number of elements in each segment as input data. The elements on the mean line are then determined corresponding to the elements on the airfoil chord. Similarly the collocation points on the airfoil surface are located corresponding to the mid-points of the elements on the airfoil chord.

On each element of the descretized mean line, the source strength is assumed to be uniform while the vorticity is assumed to have a linear variation. Following Basu [104], two point sources are placed on either end of the last mean line element nearest to the leading edge, while one point vortex is placed at the mid-point of the same element, see Figure 3.6. Denoting the internal singularity contributions to the velocity field by the subscript I, we have the following sum formulae for the velocities

$$u_I(x, y) = \sum_k a^k(x, y) \, \sigma_k + \sum_k c^k(x, y) \, \gamma_k \qquad (3.4.9)$$
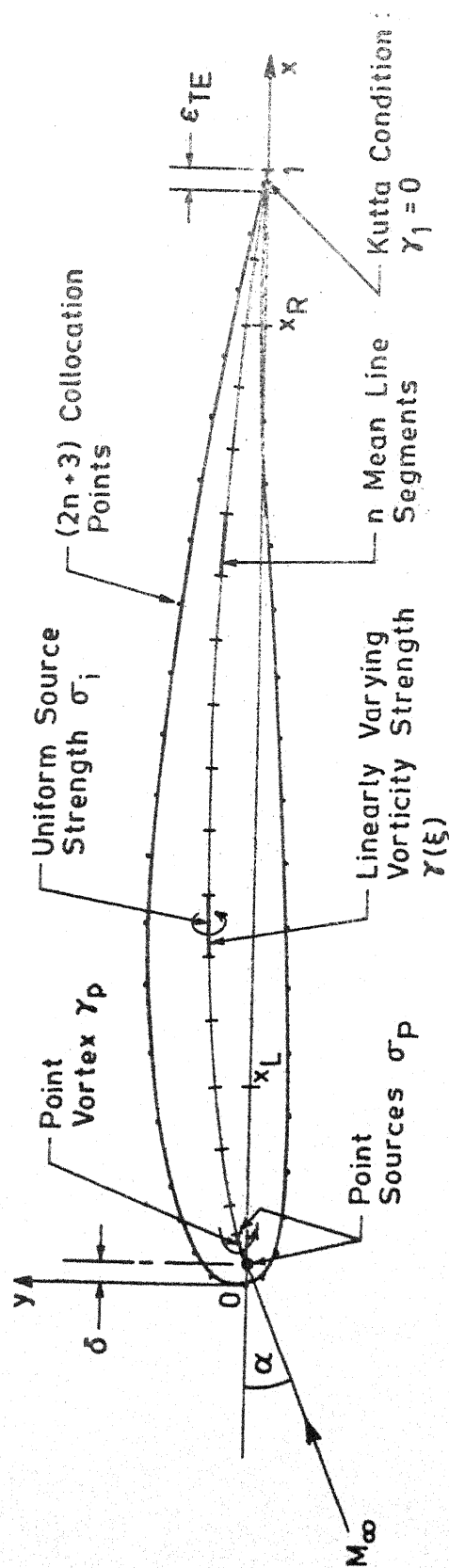
Division of Airfoil Chord into Elements

(i) $\delta \leqslant x \leqslant x_L$: Cosine Law Distribution

(ii) $x_L \leqslant x \leqslant x_R$: Uniform Distribution

(iii) $x_R \leqslant x \leqslant 1 - \varepsilon_{TE}$: Uniform Distribution

Fig. 3.6.   Discretization of the Mean Line and Choice of the Collocation Points.

$$v_I(x, y) = \sum_k b^k(x, y) \sigma_k + \sum_k d^k(x, y) \gamma_k \qquad (3.4.10)$$

where the summations are carried out over all the singularity elements constituting the unknowns. The influence co-efficients a, b, c and d for the internal singularities are given in Appendix F. Note that the contributions from the point singularities have been absorbed in the vectors σ and γ.

The perturbation velocities are therefore given by

$$u(x, y) = u_I(x, y) + u_F(x, y) \qquad (3.4.11)$$

$$v(x, y) = v_I(x, y) + v_F(x, y) \qquad (3.4.12)$$

When the mean line is divided into n segments, we have (2n + 4) unknown singularity strengths to be determined. This implies that the tangency condition must be specified at (2n + 3) collocation points, Figure 3.5, yielding as many equations. The remaining equation necessary for a unique solution is obtained from the Kutta condition, which in the present method is satisfied by setting the vorticity strength to zero at the trailing edge.

Inserting Eqs. (3.4.11) and (3.4.12) in Eq. (3.2.9), we obtain the following equations for the tangency boundary condition.

$$v_I - F'_{u,1}(x) u_I = F'_{u,1}(x) \cos\alpha - \sin\alpha + F'_{u,1}(x) u_F - v_F$$

$$\text{on} \quad Y = F_{u,1}(x) \qquad (3.4.13)$$

In matrix form Eq. (3.4.13) can be written as

$$C_{ik}[\{\sigma_k\}\{\gamma_k\}]^T = [N_i]^T + [h_i]^T \qquad (3.4.14)$$

where

$$N_i = F'_{u,1}(x_i) \cos\alpha - \sin\alpha \qquad (3.4.15)$$

$$h_i = F'_{u,1}(x_i) \, u_F(x_i, y_i) - v_F(x_i, y_i) \qquad (3.4.16)$$

and (x_i, y_i) denote the collocation points on the airfoil surface.

The vector h_i arises because of the presence of the non-linear term. Setting $h_i \equiv 0$, gives the solution for incompressible flow. The matrix C remains fixed throughout the iteration scheme and hence may be inverted and stored at the beginning of the iteration.

3.4.3. Non-linear Source Term

From Eqs. (3.2.5), (3.2.18) and (3.2.22), the field source term may be expressed as

$$\bar{g} = \frac{Q}{c^2} \frac{\partial}{\partial s} \left(\frac{Q^2}{2}\right) + T \qquad (3.4.17)$$

The first term in Eq. (3.4.17) represents the inviscid part of the field source distribution. The streamwise derivatives occurring in this term may be evaluated directly, but the following procedure for calculating them enables the governing equations to be satisfied explicitly. Since

$$\frac{\partial}{\partial x}\left(\frac{Q^2}{2}\right) = U \frac{\partial u}{\partial x} + V \frac{\partial v}{\partial x}$$

$$\frac{\partial}{\partial y}\left(\frac{Q^2}{2}\right) = U \frac{\partial u}{\partial y} + V \frac{\partial v}{\partial y}$$

we obtain

$$Q \frac{\partial}{\partial s}\left(\frac{Q^2}{2}\right) = U \frac{\partial}{\partial x}\left(\frac{Q^2}{2}\right) + V \frac{\partial}{\partial y}\left(\frac{Q^2}{2}\right) \qquad (3.4.18)$$

Hence if u_x and u_y are known, using the irrotationality condition and the governing equation we obtain

$$v_x = u_y \qquad \text{and} \qquad v_y = \bar{g}^{old} - u_x \qquad (3.4.19)$$

The derivatives of u are computed on the $\nu - \theta$ plane using second order accurate central differences at all interior points and one sided differences at all boundary points. The derivatives on the physical plane are obtained using Eqs. (3.4.2) and (3.4.6).

The second term in Eq. (3.4.17) represents the artificial viscosity. Assuming that the streamlines on the physical plane do not deviate very much from the ν-co-ordinate lines in the supersonic regions for the airfoils to be considered, we may use upwind differencing directly on the physical plane. Since the i-indexing starts from the trailing edge on the upper surface and increases continually through the leading edge to the trailing edge on the lower surface, upwind differencing amounts to forward differencing on the upper surface of the airfoil and backward differencing on the lower surface. Such a simplified differencing procedure for the artificial viscosity term works satisfactorily for the airfoils with small camber. The artificial viscosity contribution may therefore be computed from

$$T_{ij} = \overline{G}_{ij} - \overline{G}_{i+1,j} \qquad \text{on the upper surface}$$

$$= \overline{G}_{ij} - \overline{G}_{i-1,j}$$

(3.4.20)

where

$$\overline{G} = -\mu \frac{\rho}{c^2} Q \frac{\partial}{\partial s} \left(\frac{Q^2}{2}\right)$$

(3.4.21)

If artificial compressibility is to be used, a similar approximation for Eq. (3.2.24) gives

$$\bar{\rho} = \rho_{ij} - \mu_{ij}(\rho_{ij} - \rho_{i+1,j}) \quad \text{on the upper surface}$$

$$\tag{3.4.22}$$

$$= \rho_{ij} - \mu_{ij}(\rho_{ij} - \rho_{i-1,j}) \quad \text{on the lower surface}$$

$\bar{g}$ is then obtained from Eq. (3.2.25).

Equations (3.4.21) and (3.4.22) often produce too little dissipation for strongly super-critical flows. Hence the switching factor μ in these equations is replaced by a factor $\lambda\mu$ where λ is simply a numerical parameter greater than one. The introduction of λ (to be henceforth called artificial viscosity parameter) provides a simple means of controlling the magnitude of dissipation for a given case and hence the stability of the iteration scheme.

3.4.4. Iterative Procedure

A simple direct iteration scheme is used to solve Eqs. (3.4.11) and (3.4.12), satisfying the boundary conditions (3.4.13) to obtain the unknown velocity field. For a purely subsonic case, the scheme represents a linear operator which, by repeatedly acting on a non-linear forcing term to satisfy the tangency boundary condition, produces quick convergence in the entire flow field in a few iterations. For super-critical cases, with the introduction of artificial viscosity, it is found that convergence can be achieved even for flow fields with fairly large embedded supersonic regions, without the addition of any artificial time damping terms. These terms have however been added in the present scheme to the field source term $\bar{g}$ to impart stability and to reduce the iteration count needed for convergence of strongly super-critical cases.

If an estimate of $\bar{g}^{(n)}$ is available at the n-th cycle, from Eqs. (3.4.11) and (3.4.12), we obtain two simple linear integral equations for the singularity distributions σ and γ which are determined by satisfying the tangency condition (3.4.13). This involves the solution of the linear system of equations (3.4.14). Once σ and γ are determined, the field source term may be recalculated from the present velocity field. This procedure is repeated till convergence is obtained in the entire flow field. Since the matrix C remains fixed throughout the iterations, it may be inverted and stored at the beginning of the iterative procedure.

The following equations describe one complete cycle of the iterative scheme. From the given $\bar{g}^{(n)}$

$$u_{F_{ij}}^{(n)} = \sum_{l} \sum_{k} e_{ij}^{kl} \bar{g}_{kl}^{(n)} \qquad \text{from Eq. (3.4.7)} \qquad (3.4.23)$$

$$v_{F_{ij}}^{(n)} = \sum_{l} \sum_{k} f_{ij}^{kl} \bar{g}_{kl}^{(n)} \qquad \text{from Eq. (3.4.8)} \qquad (3.4.24)$$

$$h_i^{(n)} = F_{u,l}'(x_i)\, u_{F_{il}}^{(n)} - v_{F_{il}}^{(n)} \qquad \text{from Eq. (3.4.16)} \qquad (3.4.25)$$

$$[\{\sigma_i\}^{(n)}\{\gamma_i\}^{(n)}]^T = [c_{ik}]^{-1}[\{N_k\}^{(n)} + \{h_k\}^{(n)}]^T$$
$$\text{from Eq. (3.4.14)} \qquad (3.4.26)$$

$$u_{ij}^{(n)} = u_{I_{ij}}^{(n)} + u_{F_{ij}}^{(n)} \qquad\qquad\qquad (3.4.27)$$

$$v_{ij}^{(n)} = v_{I_{ij}}^{(n)} + v_{F_{ij}}^{(n)} \qquad\qquad\qquad (3.4.28)$$

$\bar{g}_{ij}^{(n+1)}$ is now obtained from Eq. (3.4.17) if aritificial viscosity is to be used and Eq. (3.2.25) if aritificial compressibility is

to be used and the cycle is complete. An artificial time dependent quantity represented by the sum

$$\alpha_1 (u_{ij}^{(n)} - u_{ij}^{(n-1)}) + \alpha_2 (v_{ij}^{(n)} - v_{ij}^{(n-1)})$$

may now be added to $\bar{g}_{ij}^{(n+1)}$ as obtained above.

It is noted that in Eqs. (3.4.25) and (3.4.26), the indices i and k run only from 1 to 2n + 4, i.e. only on the grid points located on the airfoil surface where the tangency condition is satisfied. It is also emphasised that the Kutta condition at the trailing edge is satisfied by setting $\gamma_1 = 0$ at that point for every iteration.

The initial estimate of $\bar{g}$ for the present computations was to simply put $\bar{g}^{(o)} \equiv 0$, which in the present formulation then leads to the incompressible solution. Convergence criterion was specified in terms of maximum change in the value of the $\bar{g}_{ij}^{(n)}$ over all the grid points. In the actual calculations the iterative scheme was terminated when $\max_{ij} |\Delta \bar{g}| \leq 0.005$. This value corresponds to a maximum of less than 0.5 percent change in the surface pressure values.

3.5. RESULTS AND DISCUSSION

Plane, two dimensional, full-potential solutions for a few selected airfoils, obtained from the numerical scheme described above will be presented here. Both sub-critical and super-critical results are discussed, but the emphasis will be on the latter. First an outline of the program code is presented, which is followed by the results of some numerical experiments – which include, a study of the effect of neglecting the non-linear

contribution to the lateral velocity component, the effects of the number and distribution of grid points, the approximation for the field source strength and the artificial viscosity parameter λ. This is then followed by the presentation of super-critical results on symmetric and lifting airfoils. The discussion of these results is broadly confined to noting features specific to the method by comparing the present results with those of finite-difference or finite-element calculations wherever the latter are available. Lastly, some typical comparisons of the FPE solutions with the TSPE solutions obtained as in Section 2.5.5 are shown and the chapter concludes with a discussion of the performance of the iterative scheme.

3.5.1. Outline of the Program Code

The plane, two-dimensional, full-potential algorithm was basically incorporated into two independent program codes taking into consideration the fact that the curvilinear grid construction is dependent only on the airfoil geometry and is independent of the angle of attack. The grid for a given airfoil is therefore generated by a program package called GRID which also calculated the influence co-efficients for the field sources and internal singularities and outputs them on the disk as data files. These data are then input to the main program FULPOT which is coded to solve the problem for arbitrary, subsonic free-stream Mach numbers and angles of attack. Starting from any given Mach number and angle of attack, a series of cases could be run, if desired, obtaining converged solutions for each case and proceeding on to the next case by changing

either the free-stream Mach number or the angle of attack or
both and taking the previously converged solution as the input
solution for the new case.

Programs were run on the computer system DEC 1090,
which has a core memory of 256 K words of which only 100 K are
available to users on the interactive time-sharing mode or on
the batch processing mode. Programs requiring larger core may
also be loaded, but the need to avail of virtual memory drasti-
cally increases the central processor (CPU) time since swapping
time is also included in the latter. Most of the present
calculations were carried out interactively on the time sharing
mode.

3.5.2. Some Numerical Experiments

(a) Effect of the Approximation Made in the Calculation of
v-component of the Perturbation Velocity

Because of the limitation on the core memory available
to users on our computer, it became necessary to incorporate an
approximation in the calculation of the v-component of the
perturbation velocity. The non-linear contribution to the v-
component of velocity represented by the second term in Eq.
(3.4.12) was calculated only for the first level of grid points
given by $j = 1$. The field influence co-efficients necessary
for the computation of this contribution was thus cut down by
about 80 percent. As a result, for super-critical flows past
lifting airfoils, the core requirement for a typical 47 x 5 grid
was about 120 K words which, though still above the 100 K core
limit set for users, was found to give reasonably speedy

executions considering that the use of virtual memory involved swapping the object program in and out of the core at the end of almost every iteration. The effect of this approximation on the surface pressure distribution was tested for a symmetric sub-critical case of flow past a NACA 0012 airfoil at $M_\infty = 0.72$, $\alpha = 0$ and the results are shown in Figure 3.7. Surface pressure distributions have been plotted for three cases with the $v_{F_{ij}}$ contribution to v_{ij} in Eq. (3.4.12) being calculated for (a) $j = 1$ level only, (b) for all the j-levels, (c) for none of the j-levels respectively. The result from Lock [105] is also plotted in the same figure for comparison. It is seen that the approximation made above, represented by case (a) results in surface pressure values which practically coincide with the unapproximated results of case (b) and both are in close agreement with the results of Lock. Further it is noted that calculation of this contribution on the surface points is essential, as otherwise the error in the pressure peak is as much as 12 percent.

The above experiment was repeated for lifting sub-critical flow past NACA 0012 airfoil at $M_\infty = 0.63$ and $\alpha = 2°$ and the results obtained were found to be similar. Though no such checks were carried out for super-critical lifting cases, it is believed that for the relatively thin and low cambered airfoils for which computations were carried out, the above approximation does not cause serious errors in the resulting pressure distributions. This conclusion is also strengthened by the good agreements achieved in some of the calculated

$M_\infty = 0.72$ $\alpha = 0°$ grid : 23×6   NACA 0012

——— v_F computed for $j = 1$ only
△ v_F computed for all j
— — — v_F neglected for all j
○   R.C. Lock [105]
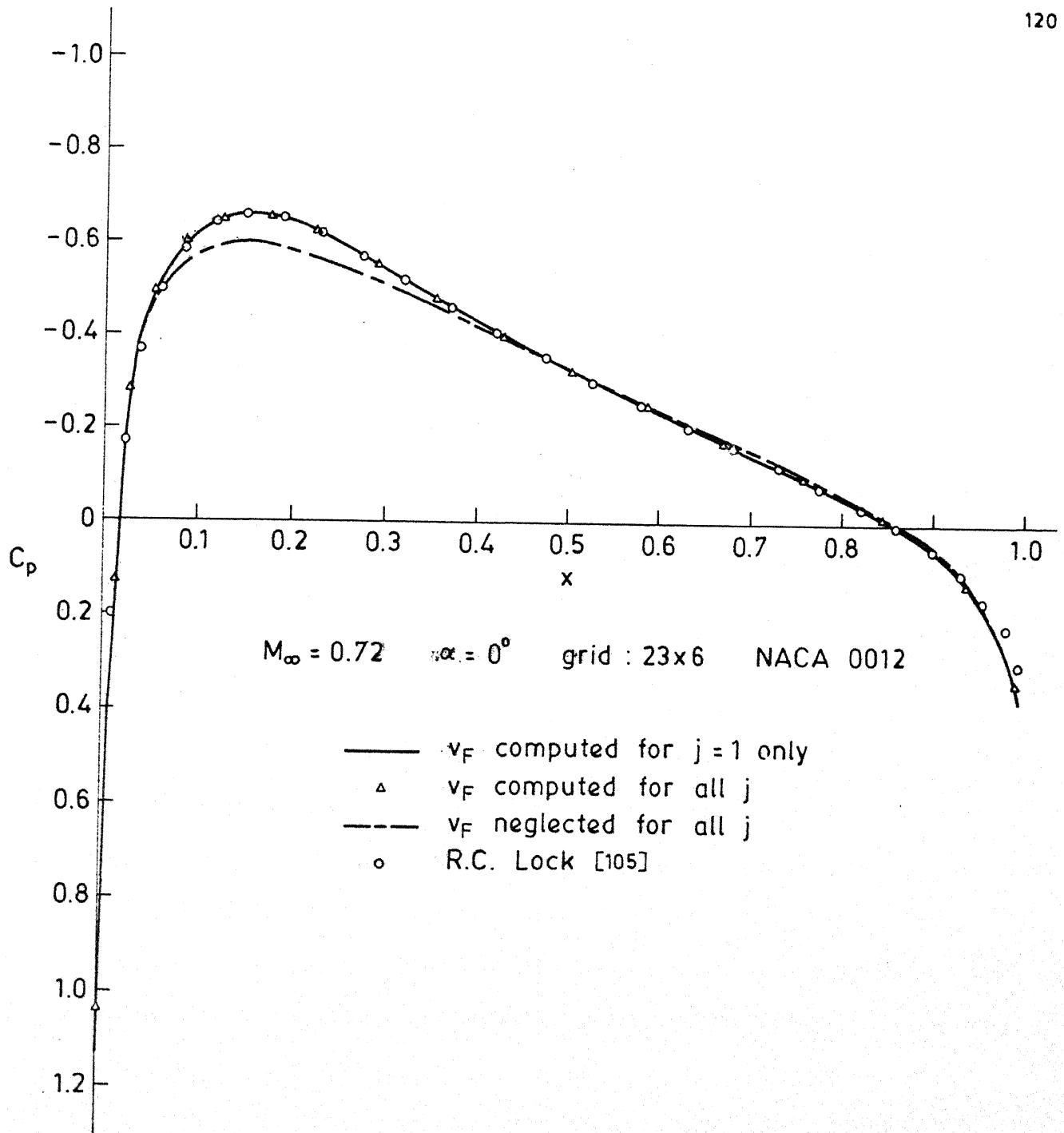
Fig. 3.7. Effect of Neglecting Nonlinear Contribution v_F to the v – Component of Velocity.

results with those obtained by finite-difference or other methods, as we shall see later. All the results that follow were obtained for the case (a) approximation as mentioned above.

(b) Grid Effects

Figure 3.8 shows the effect of a slight variation in the chord-wise distribution of the grid points on the surface pressure for a NLR-0.11-0.75-0.90, quasi-elliptical airfoil section. It is seen that small changes in the grid point distribution on the chord have almost no effect on the surface pressure distribution.

Figure 3.9 shows comparisons of the sub-critical pressure distribution on the NACA 0012 airfoil obtained by using (33 x 5) and (41 x 6) grids, with grid extent remaining the same. Lock's test data for this case are also plotted in the figure. Grid refinement is seen to result in higher peaks on the upper surface of the airfoil near the leading edge. The upper surface pressure distribution in the interval $0.1 \leq x \leq 0.9$ remains practically unaffected by grid refinement. On the lower surface also, grid refinement had little effect on the computed pressure distribution which is seen to agree well with Lock's results except for a small discrepancy near the trailing edge. This disagreement is believed to be due to the fact that the results of Lock are for an airfoil which was extended slightly beyond the trailing edge at 1.0 so as to make it closed, whereas no such extension was made at the trailing edge in the present calculations.
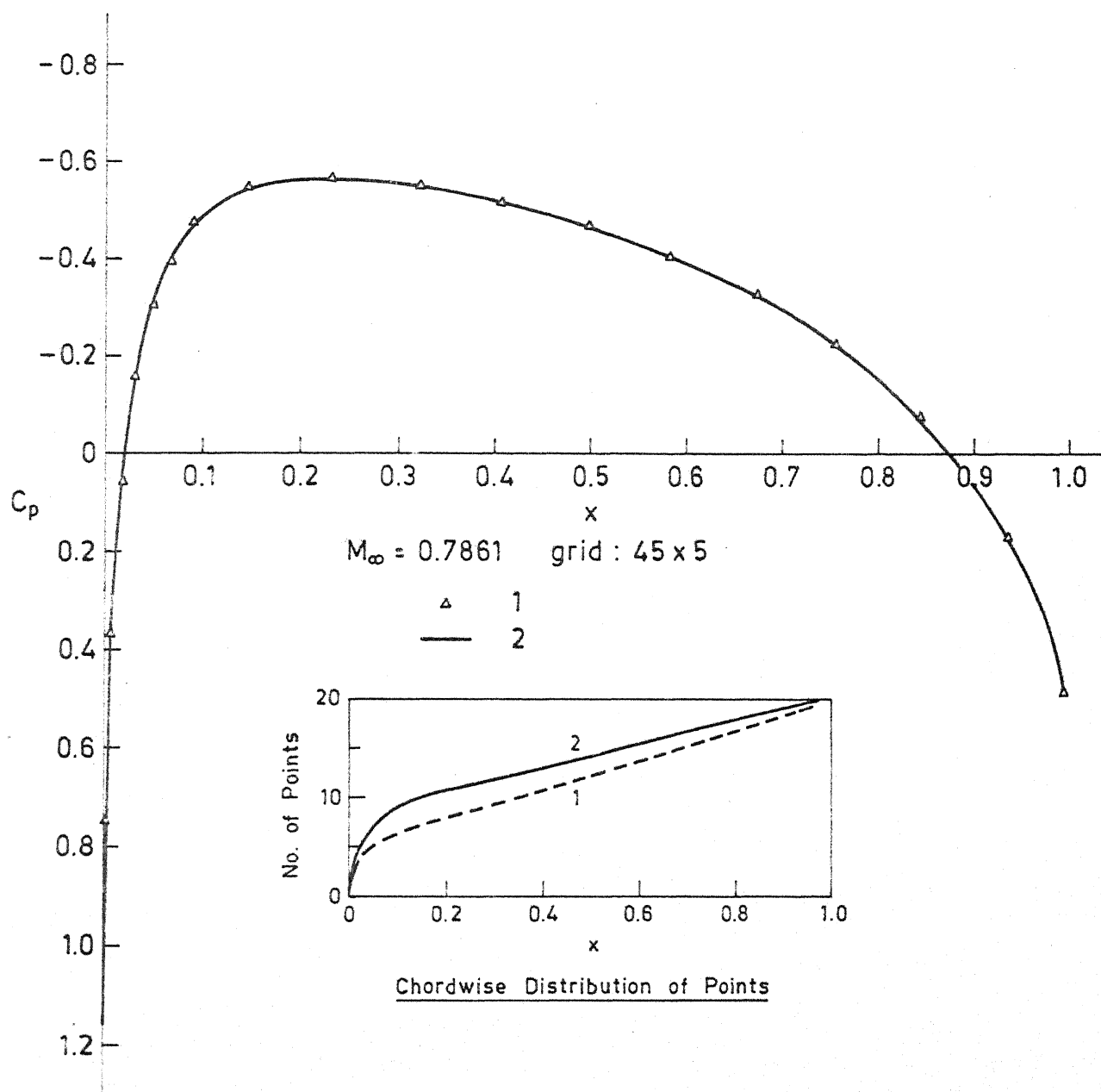
Fig. 3.8. Effect of Chordwise Grid Point Distribution on Computed Surface Pressure for NLR - 0.11- 0.75- 0.90 Airfoil.

$M_\infty = 0.63$ $\alpha = 2°$

	grid	C_L
o,÷	33 x 5	0.328
△,▽	41 x 6	0.329
——	R.C. Lock	0.335
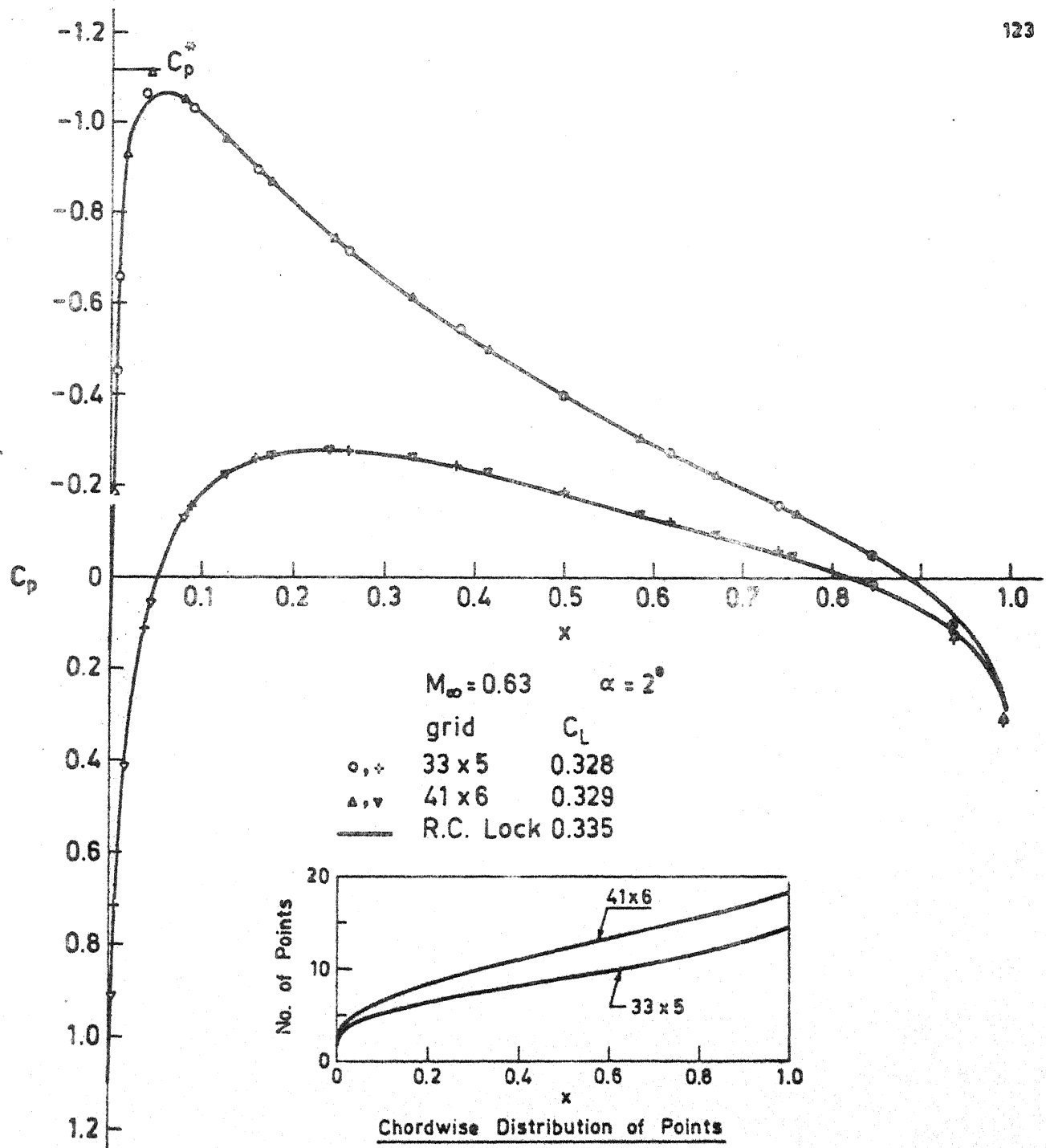
Fig. 3.9. Comparison of Coarse and Fine Grid Results : Subcritical Surface Pressure Distribution on the NACA 0012 Airfoil.

Table 3.1

Surface Pressure Distribution for Sub-critical Lifting Flow Past NACA 0012 Airfoil

$M_\infty = 0.63$ $\alpha = 2°$ Grid: 33 x 5 $C_L = 0.328$

x	C_p Upper	Lower
0	0.9481	0.9481
0.0005	0.5417	1.1013
0.007	0.3578	0.7961
0.03	-1.0184	0.2063
0.075	-1.0644	-0.1059
0.139	-0.9370	-0.2373
0.217	-0.7903	-0.2781
0.305	-0.6478	-0.2697
0.40	-0.5144	-0.2339
0.503	-0.3921	-0.1845
0.609	-0.2832	-0.1303
0.715	-0.1834	-0.0731
0.821	-0.0773	-0.0037
0.927	0.0865	0.1216
0.99	0.3128	0.3202

125

For super-critical flows, it was essential to keep the grid point spacing in the supersonic region approximately uniform in the x-direction so as to ensure that the dissipation term remained proportional to the artificial viscosity function with a fixed Δx as the constant of proportionality in this region. Since the airfoils considered are relatively thin and have small camber, the artificial viscosity computed from Eq. (3.4.20) or (3.4.22) satisfies this requirement if the chordal step size Δx is uniform. The interval $x_L < x < x_R$ shown in Figure 3.6 denotes the anticipated extent of the supersonic region for a given flow. To capture reasonably sharp shocks, Δx was fixed at 0.05 in this interval. With this choice, a typical grid used for symmetric cases was 26 x 6 and for lifting cases, 51 x 5. On the physical plane, the latter gave a grid extent of $-0.75 \leq x \leq 1.30$ and $-1.44 \leq y \leq 1.44$. For sub-critical cases a coarser 33 x 5 grid was generally used with the grid extent remaining fixed. A typical grid constructed for a NACA 64A410 airfoil is shown in Figure 3.5.

(c) Field Source Approximation

In all the above calculations, the field source strength for all field panels, except those that are adjacent to the airfoil surface, i.e. $j = 1$ row, was computed at the centre of the panel. For the field panels adjacent to the airfoil surface two different approximations for $\bar{g}$ were tried. For scheme A, the computed values of $\bar{g}$ at the collocation points were directly used. The results presented above are those of scheme A. For

scheme B, a linearly interpolated value between the values at
j = 1 and j = 2 was used.

Figure 3.10 shows for the quasi-elliptical airfoil
section NLR-0.11-0.75-0.90 a comparison of the surface pressure
distributions obtained by the two schemes, with Lock's sub-
critical results. The computed pressure distributions show
disagreement with Lock's results near the leading edge region.
A similar discrepancy was noticed earlier for the NACA 0012
airfoil, Figure 3.9. For the NLR profile, Figure 3.10 shows
that scheme B results in better agreement in the region ahead of
the peak although the peak velocities are considerably smaller
than those calculated by Lock. Scheme A calculations produce
higher peak velocities but the agreement with Lock's results in
the region between the leading edge and the velocity peak is
poor.

Different ways of evaluating the field source term
were tried to achieve better agreement between the calculated
values and Lock's results in the region between the leading edge
and the velocity peak, but they proved unsatisfactory. One of
the alternatives tried was, the calculation of $\bar{g}$ in terms of
the density ρ and its derivatives in the form $\bar{g} = -Q \frac{\partial}{\partial s} (\ln \rho)$
but this did not qualitatively alter the results. Similarly
different distributions of grid points near the leading edge
region also did not affect the results. Though the reasons
for the discrepancy could not be clearly identified, it is
believed to be due to the following. Primarily, the constant
source approximation made for the field panels in the leading
edge region, where the gradients are very large, is a source
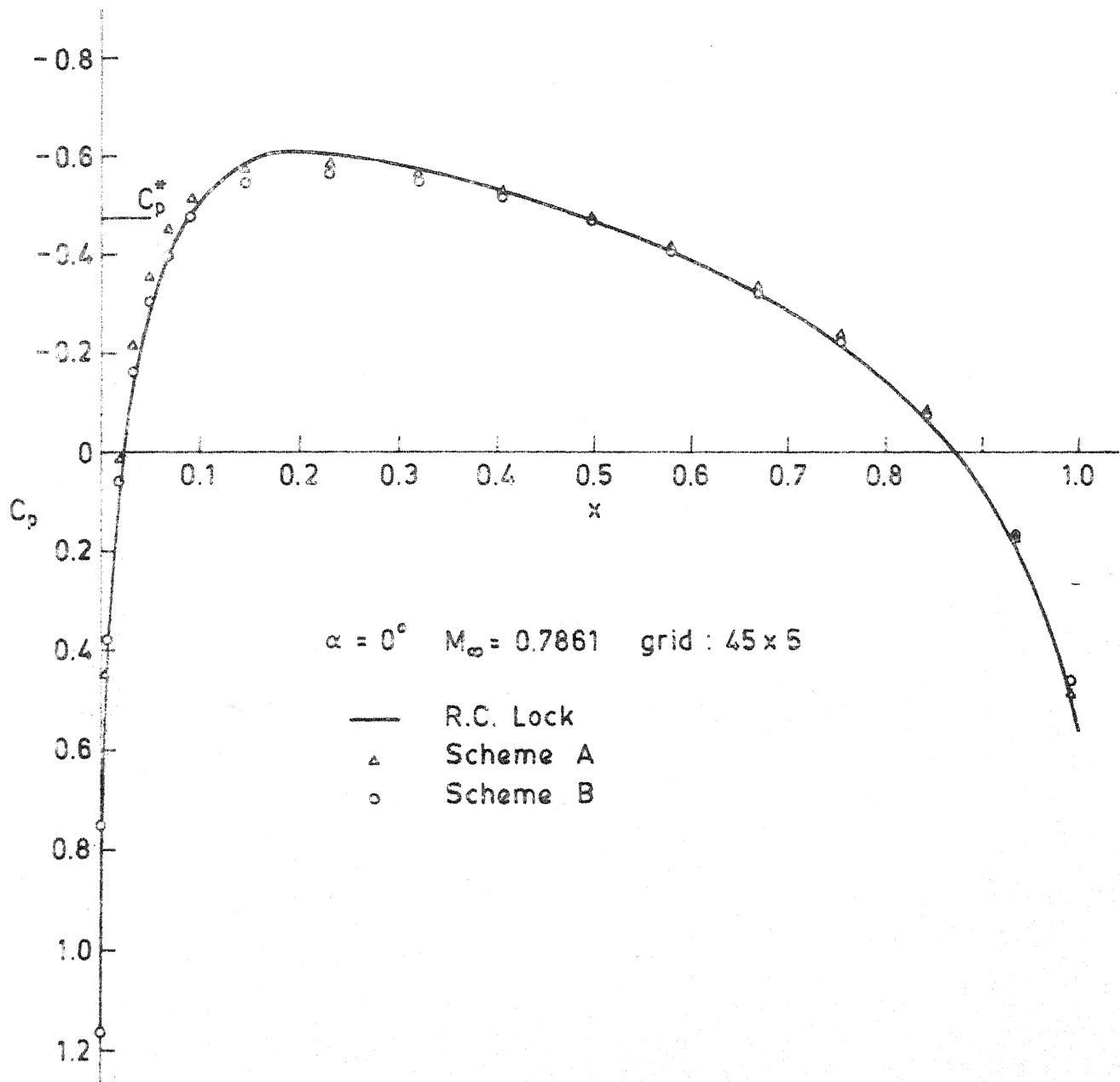
Fig. 3.10 Surface Pressure Distribution on the Quasi-Elliptical Airfoil
NLR-0.11-0.75-0.90.

Table 3.2

Co-ordinates for the Quasi-Elliptical Airfoil NLR-0.11-0.75-0.90

x	y	Curvature = $d\theta/ds$
-1.79504	0.00000	49.100807
-1.79307	0.00918	30.8
-1.79046	0.01432	21.5
-1.78714	0.01920	15.79
-1.78303	0.02414	12.01
-1.77798	0.02928	9.33
-1.77184	0.03470	7.34
-1.76437	0.04048	5.822
-1.75528	0.04668	4.630
-1.74418	0.05340	3.681
-1.73052	0.06073	2.916
-1.71360	0.06873	2.295
-1.69241	0.07756	1.7910
-1.66555	0.08733	1.3831
-1.63114	0.09815	1.05582
-1.58661	0.11009	0.79811
-1.52848	0.12310	0.59896
-1.47946	0.13236	0.49208
-1.45077	0.13718	0.44399
-1.39976	0.14485	0.37417
-1.32987	0.15376	0.297747
-1.29638	0.15748	0.267000
-1.24843	0.16228	0.229433
-0.94118	0.18268	0.120369
-0.83289	0.18680	0.106563
-0.73948	0.18933	0.098506
-0.65319	0.19090	0.093190
-0.57171	0.19175	0.089533
-0.49395	0.19200	0.086994
-0.41934	0.19175	0.085257
-0.34754	0.19106	0.084117
-0.27837	0.18998	0.083436
-0.14708	0.18684	0.083092
-0.05444	0.18376	0.083498
0.03372	0.18016	0.084339
0.09011	0.17752	0.085101
0.14468	0.17470	0.086004
0.19748	0.17173	0.087030
0.24861	0.16861	0.088171
0.29799	0.16539	0.089412
0.34589	0.16205	0.090744
0.39217	0.15863	0.092156

Table 3.2 (Continued)

x	y	Curvature = dθ/ds
0.45876	0.15335	0.094401
0.52204	0.14795	0.096768
0.58223	0.14244	0.099225
0.63943	0.13687	0.101737
0.69383	0.13126	0.1043
0.74555	0.12563	0.1068
0.79471	0.12002	0.1092
0.84152	0.11442	0.112
0.90033	0.10703	0.116
0.95545	0.09974	0.115
1.00698	0.09258	0.117
1.07833	0.08214	0.120
1.14317	0.07210	0.118
1.20234	0.06249	0.113
1.25596	0.05332	0.111
1.35001	0.03656	0.0520
1.42858	0.02234	-0.070293
1.49459	0.01079	-0.37512
1.52352	0.00617	-0.7015
1.55001	0.00250	-1.3626
1.58224	0.00000	

Table 3.3

Symmetric Surface Pressure Distribution Over Quasi-Elliptical
Airfoil NLR-0.11-0.75-0.90

$M_\infty = 0.7861$ $\alpha = 0°$ Grid: 45 x 5

x	C_p
0	1.16
0.0005	1.0516
0.0049	0.5213
0.0193	-0.0808
0.0461	-0.3685
0.0811	-0.5065
0.135	-0.5751
0.205	-0.5891
0.275	-0.5734
0.345	-0.5494
0.415	-0.5182
0.485	-0.4797
0.555	-0.4339
0.625	-0.3789
0.695	-0.3105
0.765	-0.2233
0.830	-0.1140
0.89	0.0350
0.95	0.2480
0.99	0.4752

of inaccuracies. Secondly, the internal singularity model
adopted in the present method appears to be increasingly sensi-
tive to the choice of δ, the leading edge gap, as the airfoil
becomes relatively more blunt. This is illustrated in Figure
3.11 which displays the computed results for the quasi-elliptical
airfoil section NLR-0.1025-0.675-1.375 at $M_\infty = 0.7557$, $\alpha = 0°$
for two different values of $\delta = 0.1 \ r_t$ and $0.2 \ r_t$. This is a
shock free super-critical case. The airfoil has a very blunt
leading edge with $r_t = 0.1665$. The figure shows that the
surface pressure distribution is erratic near the leading edge,
changing rather drastically in this region as δ is increased
from $0.1 \ r_t$ to $0.2 \ r_t$. The pressure curve in the interval
$0.2 \leq x \leq 1.0$ is quite smooth though the computed values
remain sensitive to δ till about 0.4 of the airfoil chord.
Thus the model gives inaccurate solutions for airfoils with
relatively large nose radii, and actually fails when the nose
radius is of $0(0.1)$ or more. Thus the disagreement between
the computed results and Lock's results is somewhat greater for
the NLR-0.11-0.75-0.90 airfoil ($r_t = 0.0204$) because it is
relatively more blunt than the NACA 0012 airfoil ($r_t = 0.0156$).

Results from schemes A and B were also compared with
the results of finite-difference calculations for a super-
critical lifting case. These comparisons are shown in Figures
3.12 and 3.22. It is clear that scheme A results show much
better over all agreement with the conservative finite-difference
results than scheme B. For this reason, the results to be
presented were all calculated using only scheme A.
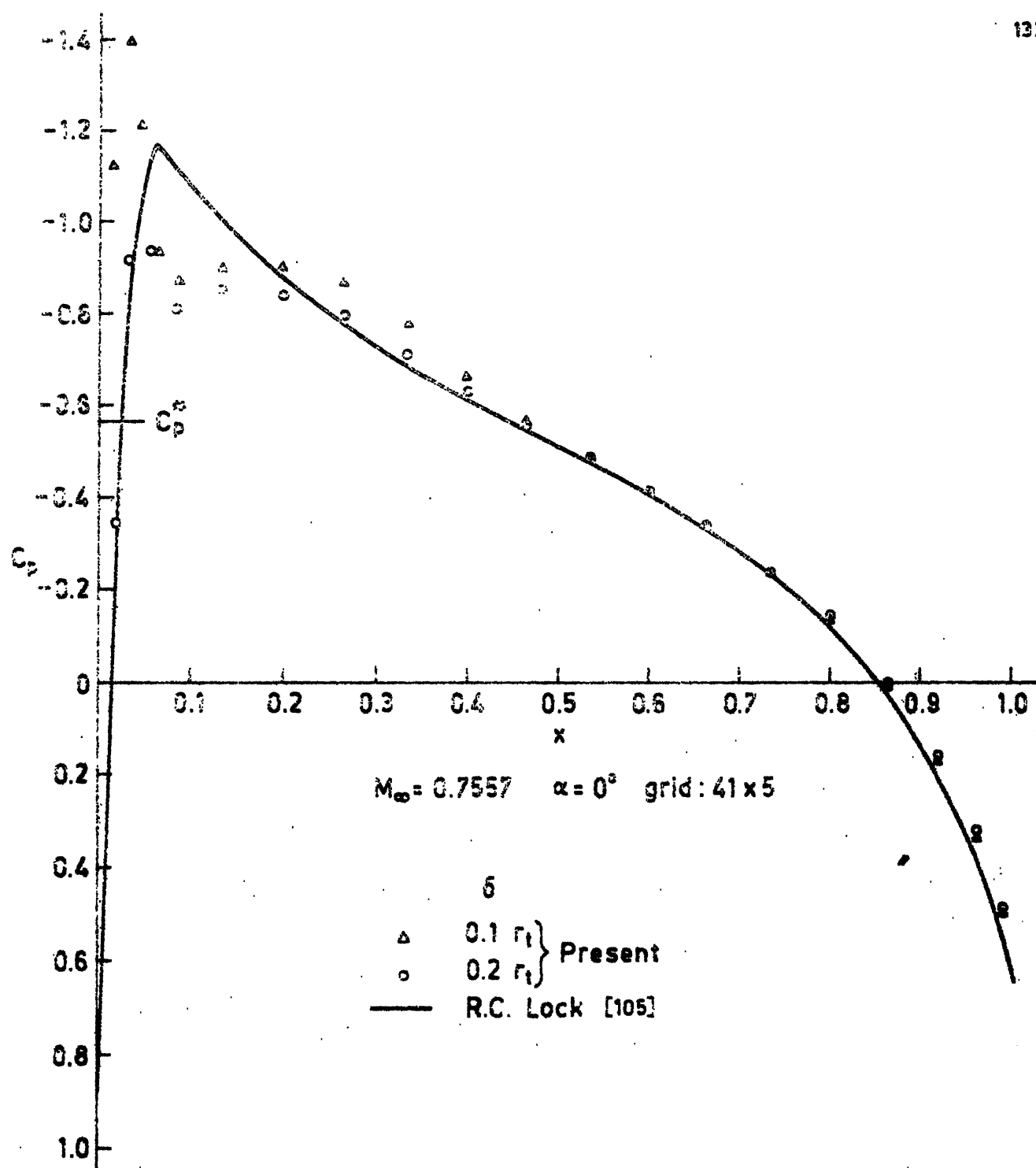
Fig. 3.11. Effect of δ : Surface Pressure Distribution on Quasi Elliptical Airfoil NLR - 0.1025 - 0.675 - 1.375.

Table 3.4

Co-ordinates for the Quasi-Elliptical Airfoil
NLR-0.1025-0.6750-1.3750

x	y	Curvature = dθ/ds
-1.69058	0.00000	5.962550
-1.68599	0.03897	5.95
-1.68141	0.05473	5.94
-1.67683	0.06656	5.94
-1.67225	0.07630	5.94
-1.66767	0.08469	5.941
-1.66310	0.09208	5.943
-1.65851	0.09873	5.944
-1.65391	0.10477	5.940
-1.64928	0.11033	5.929
-1.64460	0.11550	5.904
-1.63984	0.12034	5.858
-1.63497	0.12492	5.787
-1.62993	0.12929	5.671
-1.62464	0.13355	5.509
-1.61905	0.13771	5.282
-1.61286	0.14195	4.991
-1.60840	0.14481	4.760
-1.60605	0.14625	4.633
-1.59830	0.15072	4.225
-1.58935	0.15542	3.797
-1.57912	0.16027	3.402
-1.56790	0.16505	3.1064
-1.55664	0.16935	2.992
-1.54660	0.17279	3.111
-1.53798	0.17546	3.130
-1.52414	0.17926	1.6930
-1.48295	0.18874	0.75233
-1.40363	0.20334	0.44926
-1.31607	0.21597	0.32544
-1.26589	0.22200	0.284110
-1.21107	0.22773	0.251461
-1.15159	0.23310	0.225478
-1.08781	0.23797	0.204587
-1.02005	0.24223	0.187383
-0.94843	0.24579	0.172493
-0.89843	0.24774	0.163222
-0.84606	0.24935	0.154169
-0.79100	0.25059	0.145239
-0.73255	0.25141	0.136536
-0.66998	0.25178	0.128305
-0.60268	0.25162	0.120864
-0.54889	0.25110	0.115972

Table 3.4 (Continued)

x	y	Curvature = $d\theta/ds$
−0.49222	0.25018	0.111760
−0.43282	0.24884	0.108267
−0.37098	0.24703	0.105494
−0.28542	0.24387	0.102864
−0.19766	0.23984	0.101343
−0.10880	0.23496	0.100786
−0.04222	0.23077	0.100911
0.02383	0.22618	0.101426
0.08893	0.22122	0.102308
0.15259	0.21594	0.103478
0.21513	0.21034	0.104903
0.27584	0.20451	0.106546
0.33476	0.19847	0.108370
0.39183	0.19225	0.110343
0.44699	0.18590	0.112434
0.50021	0.17944	0.114613
0.55152	0.17289	0.116848
0.60092	0.16629	0.1192
0.66387	0.15744	0.1226
0.72358	0.14859	0.1245
0.78037	0.13974	0.130
0.83405	0.13097	0.129
0.88490	0.12230	0.132
0.93307	0.11377	0.133
1.00051	0.10125	0.134
1.06262	0.08914	0.131
1.11930	0.07762	0.125
1.17250	0.06644	0.1121
1.22095	0.05598	0.0910
1.26560	0.04614	0.057322
1.34448	0.02855	−0.07781
1.41119	0.01412	−0.42141
1.44054	0.00830	−0.7959
1.46746	0.00362	−1.5593
1.50134	0.00000	
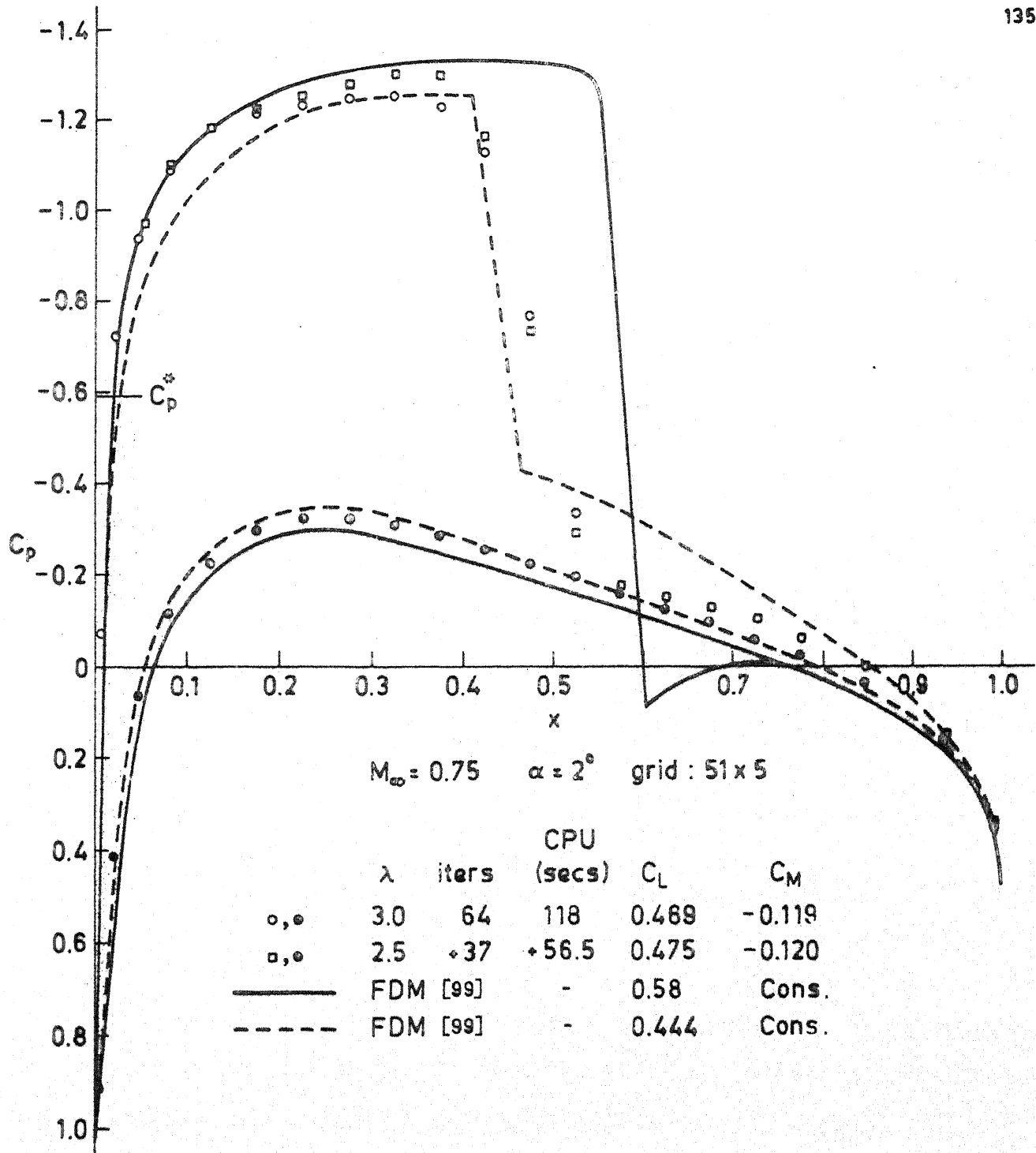
Fig. 3.12 Surface Pressure Distribution on a NACA 0012 Airfoil, Scheme B.

(d)   Artificial Viscosity Parameter

The artificial viscosity parameter λ is significant from the point of view of convergence of the iteration scheme because, once the artificial viscosity form is chosen, the parameter λ provides a simple means of controlling the magnitude of the dissipation term in the supersonic region at supercritical Mach numbers.

Figure 3.13 shows the effect of the viscosity parameter λ on the surface pressure distribution calculated for symmetric flow past a NACA 0012 airfoil at $M_\infty = 0.82$. For this case, λ was initially set at 3.0 and convergence was achieved in 50 iterations. The calculations were repeated with $\lambda = 2.6$ and 2.4. For $\lambda = 2.4$, the iterative scheme showed signs of going into an oscillation and was terminated after 100 iterations; the surface pressure distribution at this stage is plotted in Figure 3.13.

Figure 3.13 shows that the peak velocities attained just ahead of the shock increase as the value of λ decreases. The shock location remains more or less fixed at about 0.575 chord. Converged solutions could still be obtained with $\lambda = 2.5$, but with a further decrease in λ to 2.4, instabilities occurred. The surface pressure plot shows the incipient formation of a peaky region ahead of the shock. The formation of such a peaky region however does not necessarily lead to divergence. But it is a sure sign of the occurrence of instabilities which may or may not amplify. Further decrease in the value of λ however leads to quick divergence with the appearance of a prominent and growing over expansion region just ahead of the shock.
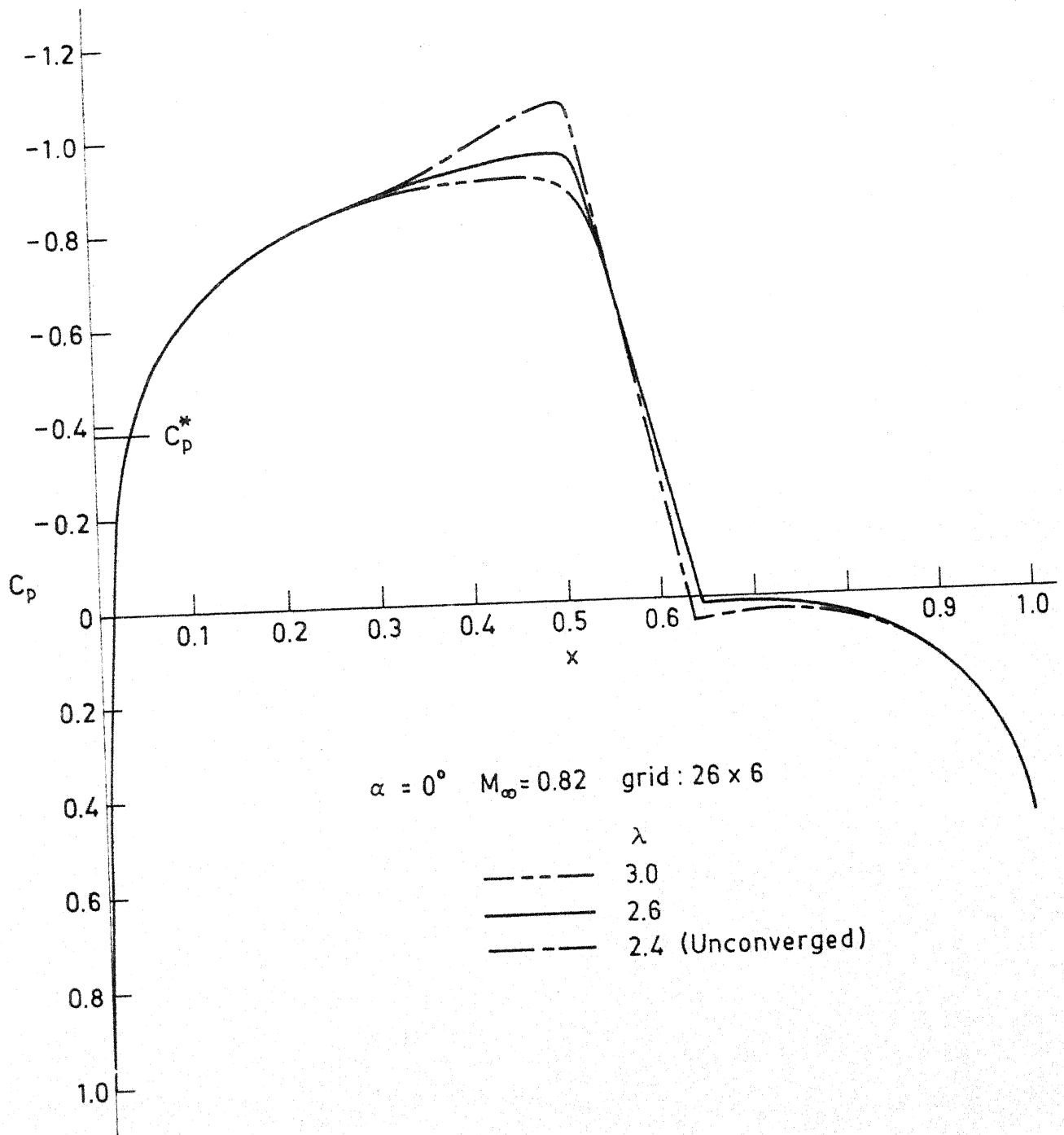
Fig. 3.13. Effect of λ on the Surface Pressure Distribution for a NACA 0012 Airfoil.
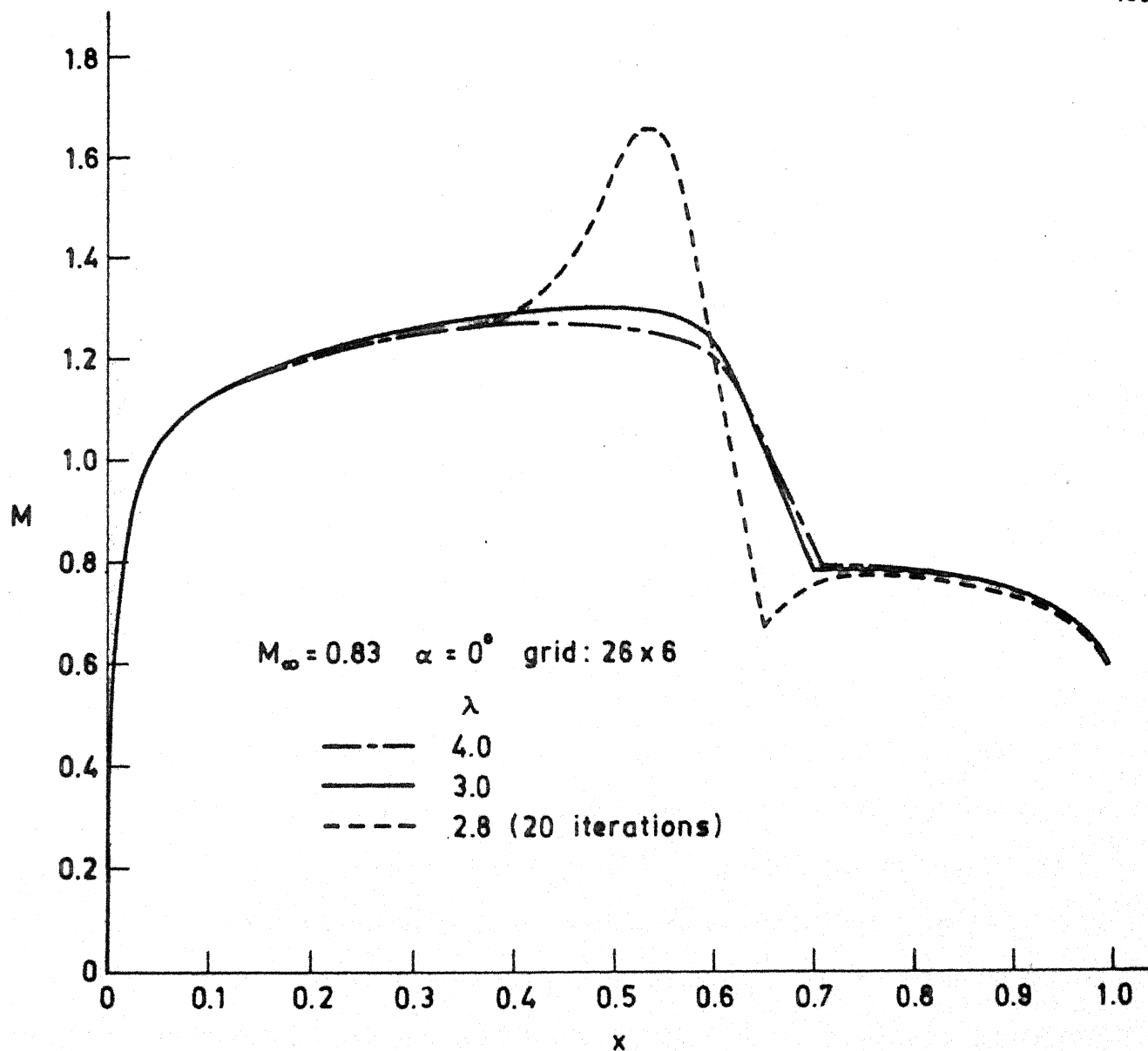
Fig. 3.14. Effect of λ on the Surface Mach Number Distribution for a NACA 0012 Airfoil.

This is clearly illustrated in Figure 3.14 in which the surface Mach number distributions are plotted for a slightly higher free-stream Mach number 0.83. The converged solutions were obtained for $\lambda = 4.0$ and 3.0. The surface Mach number distribution obtained at the end of 20 iterations for $\lambda = 2.8$ shows a sharp and prominent peak in the shock region which finally led to divergence of the scheme in about 35 iterations.

Hence it appears that solutions differing significantly in the shock region, but agreeing broadly elsewhere, are obtainable depending upon the value chosen for the artificial viscosity parameter λ. Such arbitrariness appears to be associated with other inviscid, potential flow formulations too and is the result of the need to introduce artificial viscosity in the numerical schemes, see for example [36]. Some circumspection is therefore necessary in choosing the appropriate solutions. In the present thesis the following criterion has been adopted: From among a sequence of solutions for decreasing values of λ, that solution is accepted which does not show a peaky velocity profile ahead of the shock and which corresponds to the least value of λ. Therefore, it becomes necessary to generate solutions for atleast two or three values of λ starting from some appropriate high value (depending upon the airfoil, free-stream Mach number and angle of attack) and decreasing in steps varying from 0.2 to 0.5. In the present work this has been done for almost every case and surface pressure distributions were obtained for atleast two values of λ.

3.5.3. Symmetric Super-Critical Results

Symmetric super-critical solutions with shocks were obtained for flow past a sharp nosed parabolic arc airfoil and the NACA 0012 airfoil at different free stream Mach numbers.

Figure 3.15 shows the surface pressure distribution obtained on a 10 percent thick parabolic arc airfoil at M_∞ = 0.84. Solutions obtained with λ = 4.0, 3.0 and 2.5 are compared with the results of conservative finite-difference calculations by Holst [13]. The present method is seen to predict higher velocity peaks as compared to Holst'f results, but the shock locations predicted by both methods agree well. The finite-difference shock is seen to be weaker than that obtained from the IEM.

Figures 3.16 and 3.17 show super-critical symmetric solutions for flow over a NACA 0012 airfoil at free-stream Mach numbers ranging from 0.78 to 0.84. Figure 3.16 shows the surface pressure plots while Figure 3.17 shows the corresponding Mach number plots. The λ value quoted for each free-stream Mach number is approximately the value at which converged solutions, satisfying the criterion cited in Section 3.5.2d, could be obtained. The Mach number plot shows that shocks are captured better for higher free-stream Mach numbers, these being sharper and better centred, as judged from the pressure plots of Figure 3.16, than the shocks at lower free-stream Mach numbers. Thus for instance, the shock at M_∞ = 0.78 is seen to be practically smeared out.

A comparison of the present results with those of conservative, finite-element calculations for two cases,
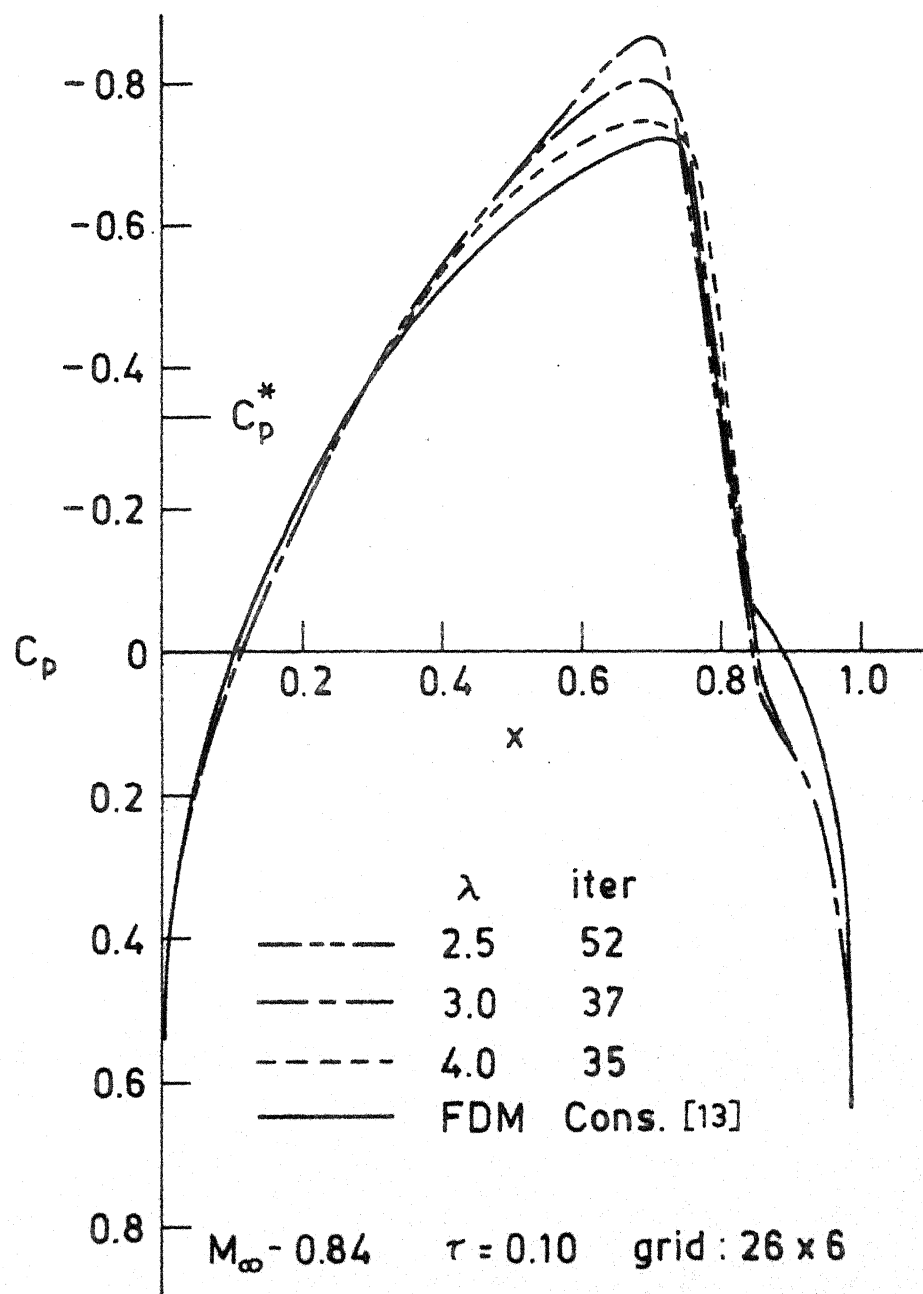
Fig. 3.15. Surface Pressure Distribution for a Parabolic Arc Airfoil.

	M_∞	λ	iters	CPU (secs)
	0.78	1.0	22	30.2
	0.80	2.0	41	43.4
	0.82	2.5	73	65.7
	0.83	3.0	75	67.5
	0.84	4.0	110	97.5

$\alpha = 0°$ grid : 26 × 6
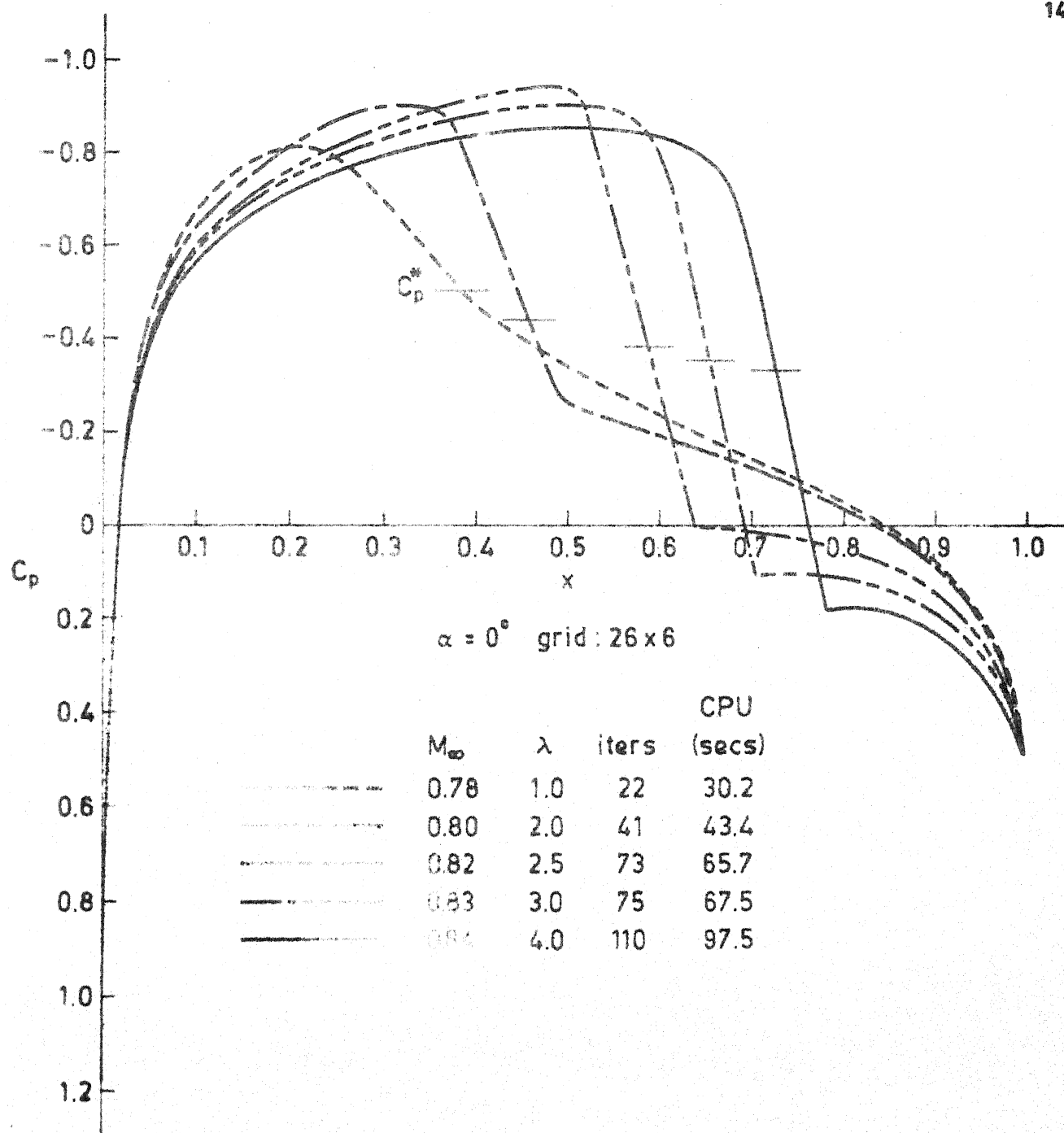
Fig. 3.16. Surface Pressure Distribution for the NACA 0012 Airfoil.
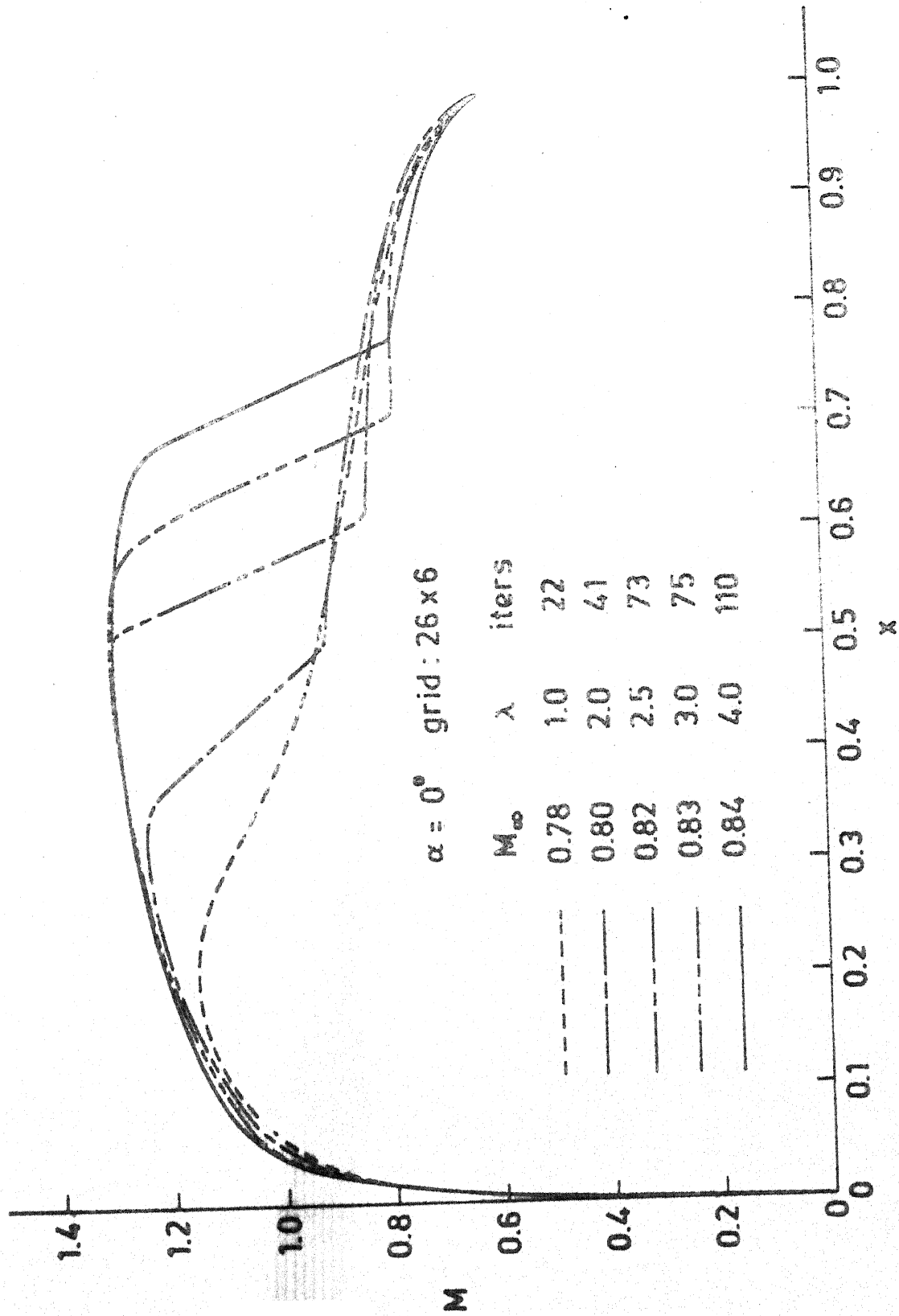
Fig. 3.17 Mach Number Distribution on the Surface of a NACA 0012 Airfoil.
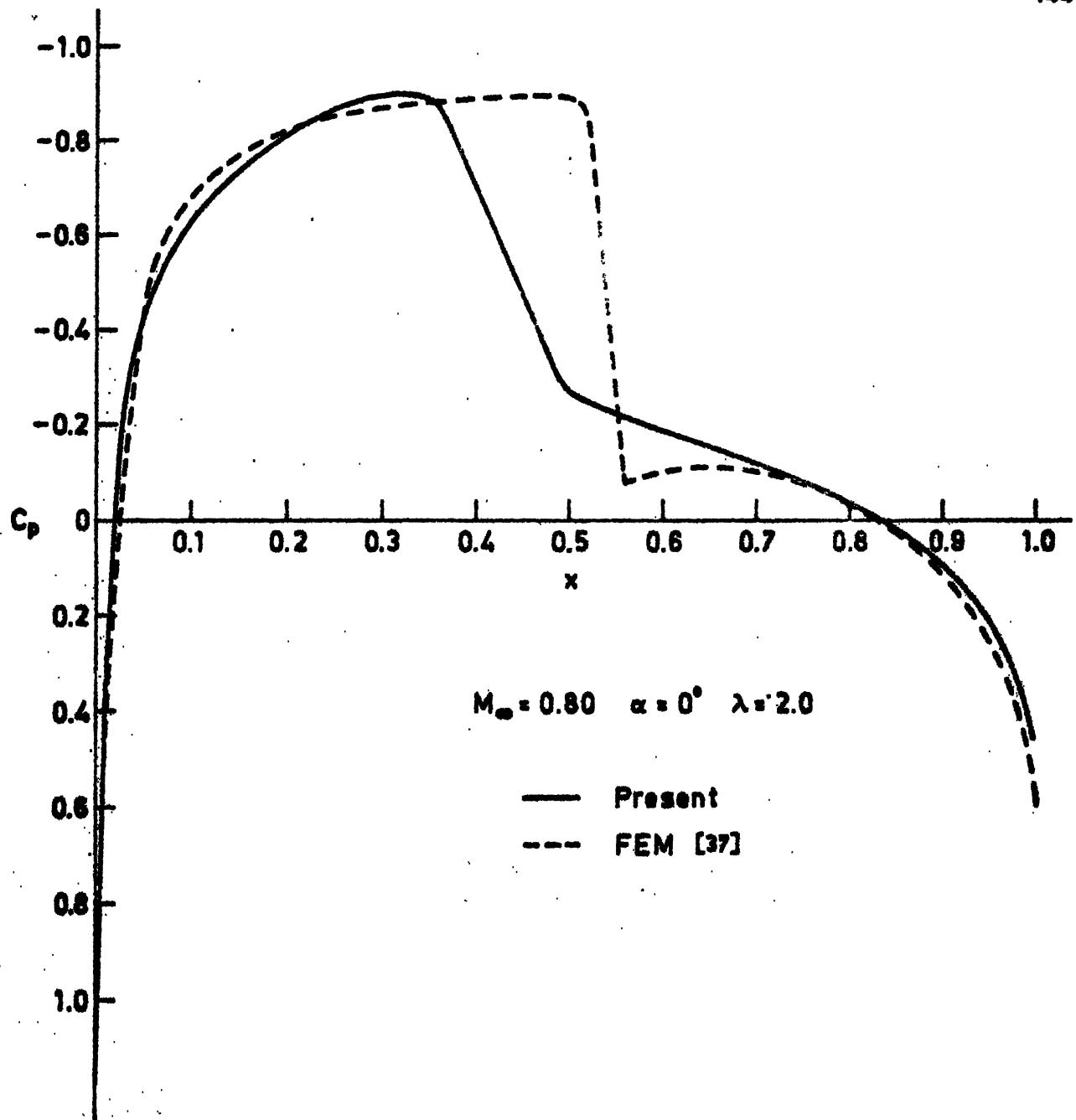
Fig. 3.18. Comparison of Surface Pressure Distribution with Finite-Element Conservative Results for a NACA 0012 Airfoil.
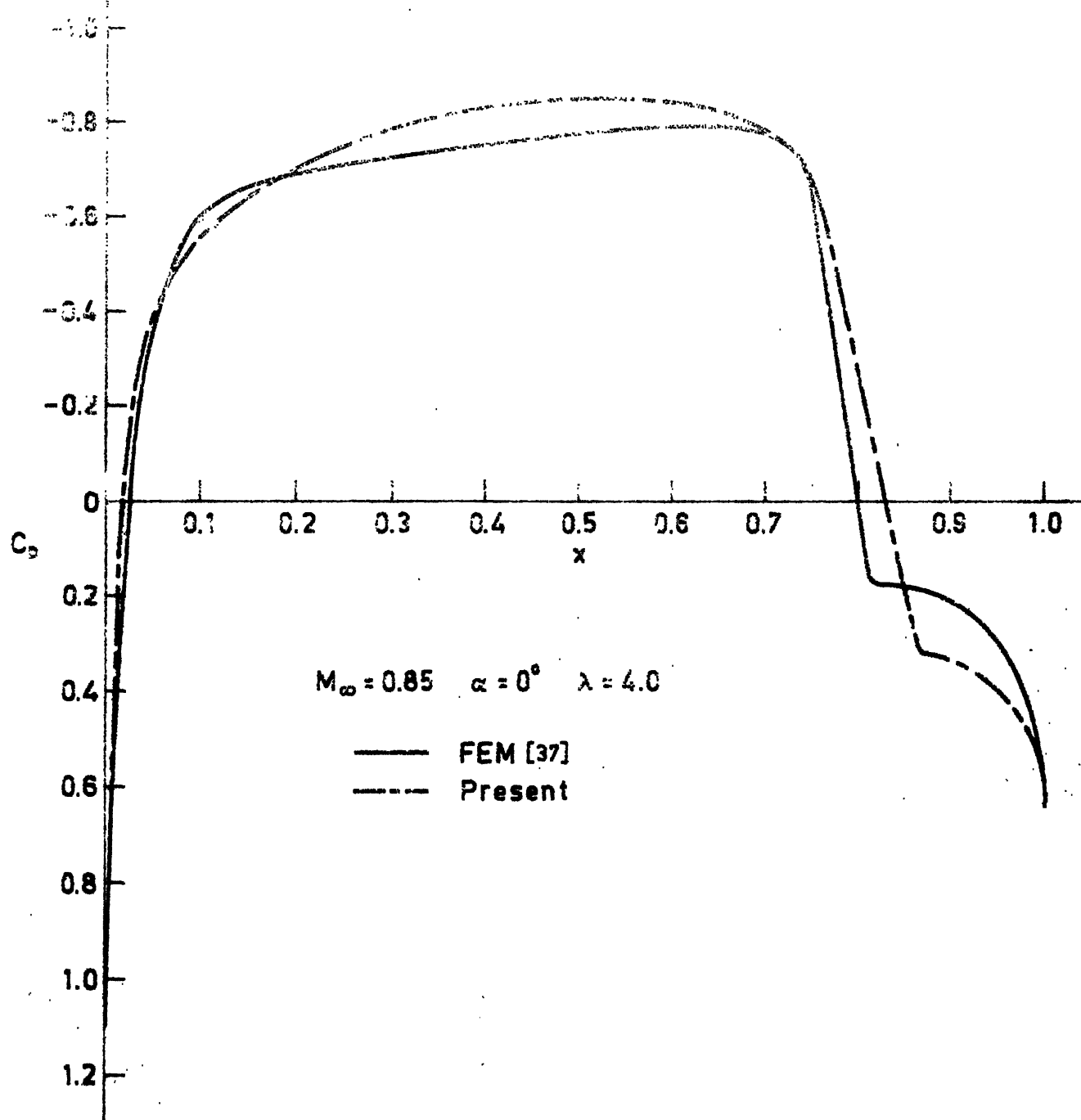
Fig. 3.19. Comparison of the Surface Pressure Distribution with Finite-Element Conservative results for a NACA 0012 Airfoil.

$M_\infty = 0.85$ $\alpha = 0^\circ$ $\lambda = 4.0$

——— FEM [37]

—·—·— Present

Table 3.5

Surface Pressure Distribution for Super-critical Symmetric
Flow Past NACA 0012 Airfoil

$M_\infty = 0.85 \quad \alpha = 0° \quad \lambda = 4.0 \quad \alpha_1 = 2.0 \quad$ Grid: 26 x 6

x	c_p
0	1.1939
0.0005	1.1078
0.0049	0.6352
0.0193	0.0173
0.0461	-0.3458
0.0811	-0.4997
0.125	-0.5981
0.175	-0.6751
0.225	-0.7264
0.275	-0.7653
0.325	-0.7950
0.375	-0.8185
0.425	-0.8356
0.475	-0.8463
0.525	-0.8512
0.575	-0.8486
0.625	-0.8367
0.675	-0.8086
0.725	-0.7529
0.775	-0.5027
0.83	0.0113
0.89	0.3281
0.95	0.3994
0.99	0.5227

illustrates the shock capturing characteristics of the present method. Figures 3.18 and 3.19 show comparisons of the surface pressure distributions obtained by the present method with the results of Bristeau [37] for $M_\infty = 0.80$ and 0.85 respectively. Figure 3.18 shows that the captured shock for the present method is rather smeared as compared to the shock obtained from the conservative, finite-element calculations. Further the shock location obtained by the present method is about two mesh widths upstream of the shock of [37]. Figure 3.19 on the other hand shows that the present shock location agrees fairly well with the one calculated by FEM although it is stronger than the latter. Similar trends were observed for the lifting cases and these results will be discussed in the next section.

Lastly it is remarked that in Figures 3.16 and 3.17, the converged results for $M_\infty = 0.84$ could be obtained only by including the artificial time-dependent term $\alpha_1 u_t$ to the field source term $\bar{g}$ with $\alpha_1 = 0.5$. The effect of α_1 on convergence history is discussed in a later section.

3.5.4. Super-Critical Results for Lifting Flows

Super-critical solutions for lifting flows were obtained for the symmetric NACA 0012 airfoil and the cambered NACA 64A410 airfoil whose profile is presented in terms of the upper and lower surface co-ordinates [109].

Figures 3.20 to 3.22 show the surface pressure distribution obtained for the NACA 0012 airfoil at angles of attack $\alpha_1 = 1.0$, 1.5 and 2.0 degrees. Values of λ for which the surface pressure distributions were obtained together with the calculated

Fig. 3.20. Surface Pressure Distribution on a NACA 0012 Airfoil.

$M_\infty = 0.75$ $\alpha = 1.5°$ grid : 51×5

		CPU		
λ	iters	(secs)	C_L	C_M
•,+ 3.0	49	95	0.35	−0.085
— 2.5	+48	+71	0.35	−0.085
- - - FDM Cons [106]				
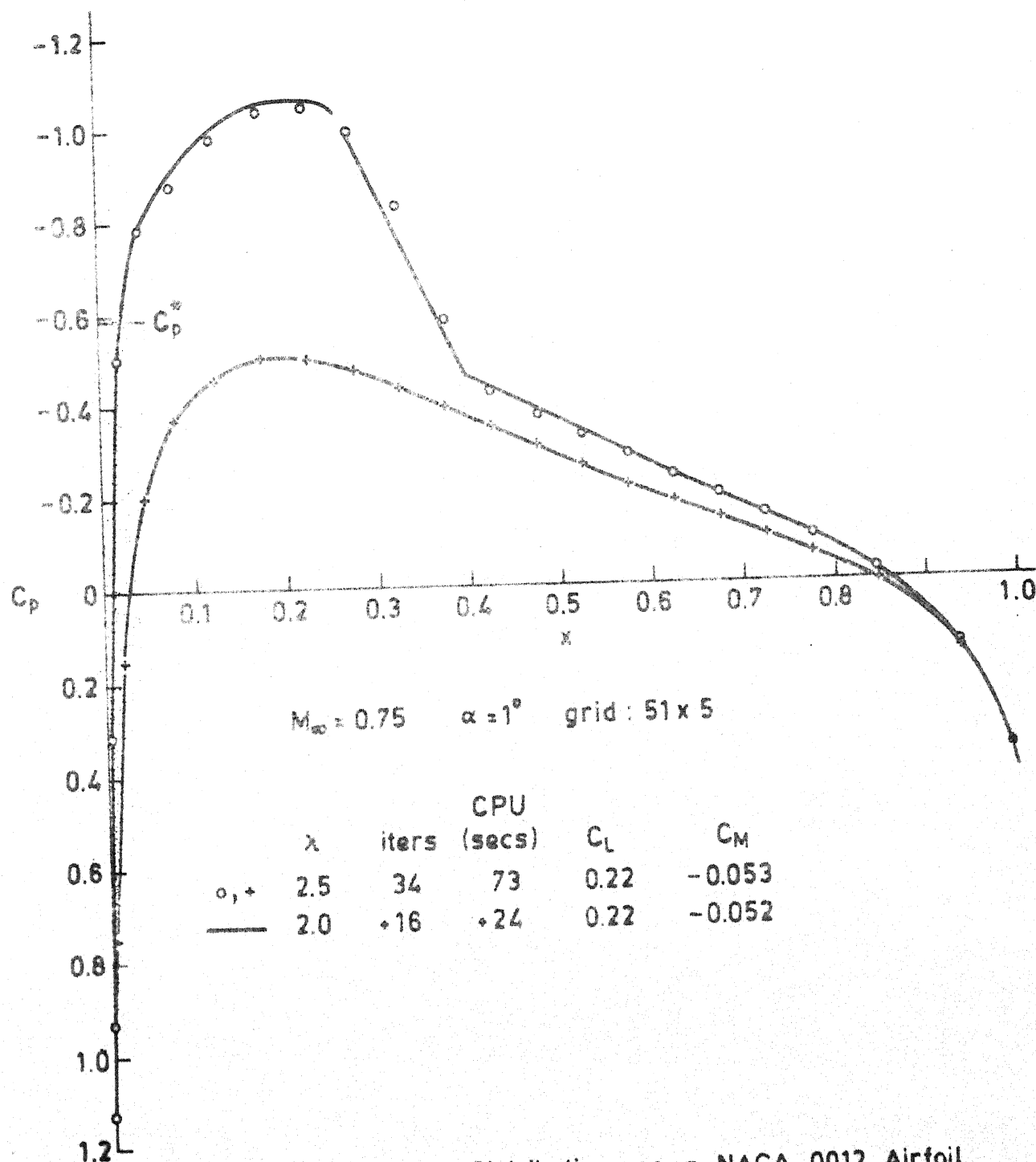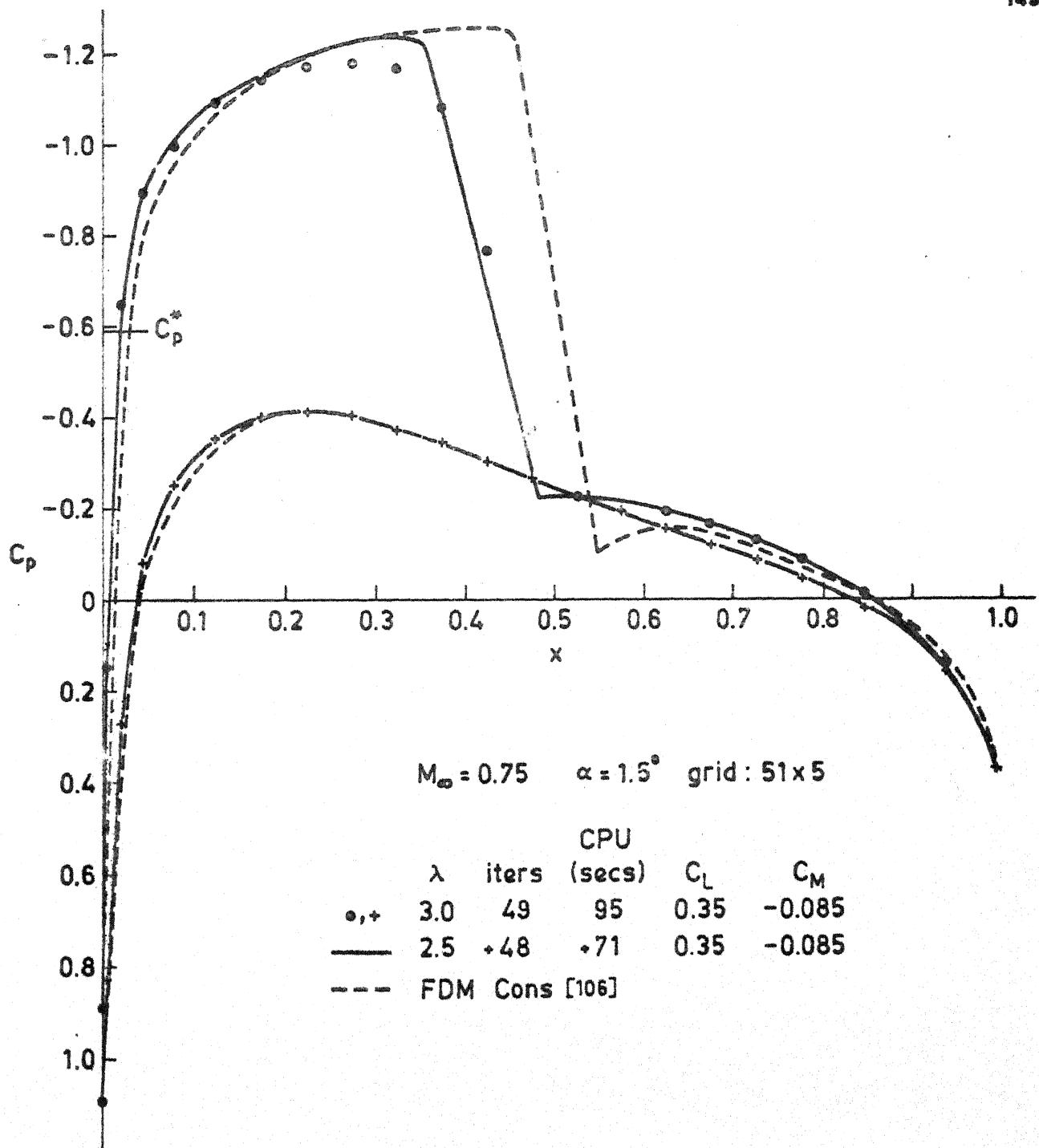
Fig. 3.21.   Surface Pressure Distribution on a NACA 0012 Airfoil
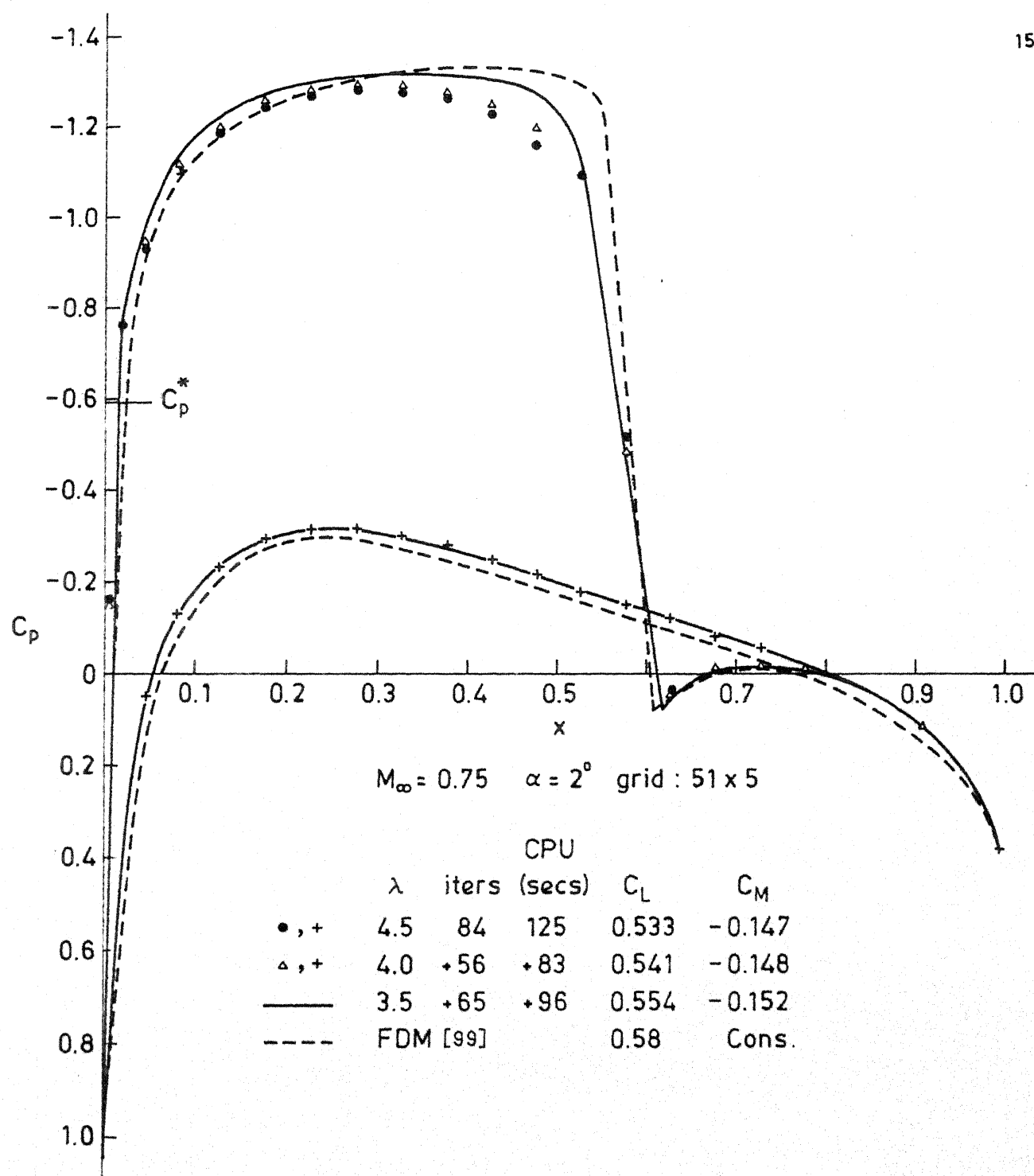
Fig. 3.22. Surface Pressure Distribution on a NACA 0012 Airfoil.

Table 3.6

Surface Pressure Distribution for Super-critical Lifting
Flow Past NACA 0012 Airfoil

$M_\infty = 0.75$ $\alpha = 2°$ $\lambda = 3.5$ $\alpha_1 = 0.5$ Grid: 51 x 5

$C_L = 0.554$ $C_M = -0.1515$ $C_D = -0.0118$

x	C_p Upper	Lower
0.0	0.9319	0.9319
0.0005	0.9276	1.1486
0.0049	-0.1361	0.9146
0.0193	-0.7713	0.4092
0.0461	-0.9633	0.0579
0.0811	-1.1299	-0.1216
0.125	-1.2140	-0.2297
0.175	-1.2708	-0.2886
0.225	-1.2978	-0.3104
0.275	-1.3109	-0.3111
0.325	-1.3156	-0.2964
0.375	-1.3128	-0.2728
0.425	-1.3032	-0.2433
0.475	-1.2654	-0.2129
0.525	-1.1320	-0.1783
0.575	-0.4677	-0.1499
0.625	0.0686	-0.1193
0.675	0.0117	-0.0808
0.725	0.0100	-0.0648
0.775	0.0044	-0.0176
0.845	0.0458	0.0414
0.935	0.1772	0.1794
0.99	0.3845	0.3843

C_L and C_M values are also indicated in the figures. The moment co-efficients were calculated about the origin. The + sign in front of a number under the column 'ifers', indicates that the iterations for the corresponding value of λ was continued from the converged solutions for the previous value of λ. Figure 3.22 shows that the computed surface pressure distribution for $M_\infty = 0.75$, $\alpha = 2°$ compares rather well with the conservative results obtained by finite-difference calculations [99] particularly as regards shock strength and location. However for $M_\infty = 0.75$ at a lower angle of attack, $\alpha = 1.5°$, Figure 3.21 shows that the computed shock is considerably weaker and is situated about one mesh width upstream of the shock calculated by another conservative finite-difference scheme [106]. Thus here again, as for the symmetric cases considered earlier, the present scheme seems to give results which show better agreement with finite-difference conservative results as the flow becomes more super-critical.

Figures 3.23 and 3.24 show the results obtained for a NACA 64A410 airfoil. Figure 3.23 shows the surface pressure distributions at three different free-stream Mach numbers, 0.735, 0.74 and 0.75, while Figure 3.24 shows the plots of the corresponding surface Mach number distributions. The rapid increase in the extent of the supersonic region and the strength of the shock as the free-stream Mach number is incre- ased from 0.735 to 0.75 is clearly evident. The Mach number plot for $M_\infty = 0.72$ is also included in Figure 3.24. This is a super-critical case and Jameson's finite-difference, quasi- linear calculations [25] show the presence of a sharpshock on
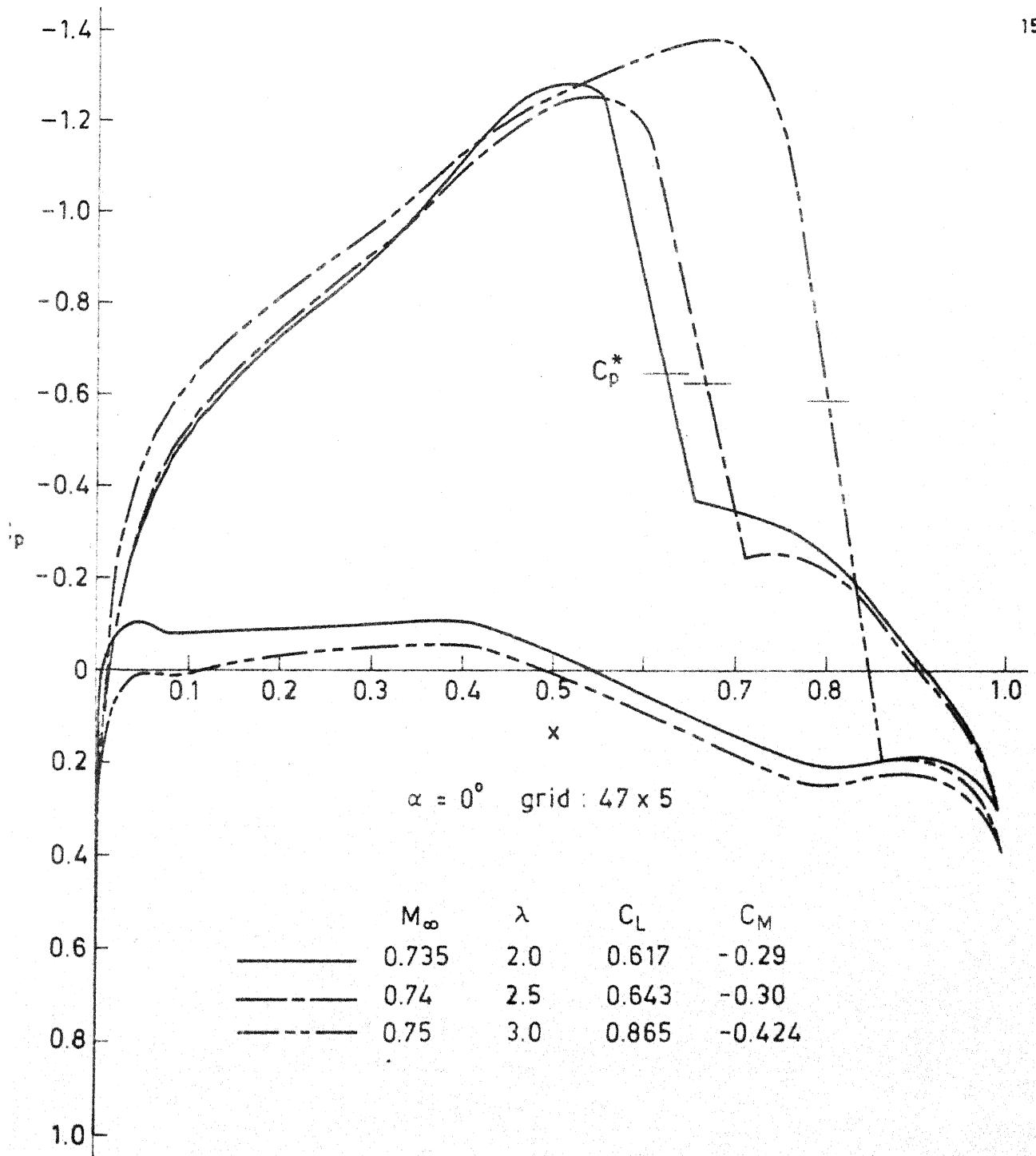
Fig. 3.23. Surface Pressure Distribution over a NACA 64A410 Airfoil.
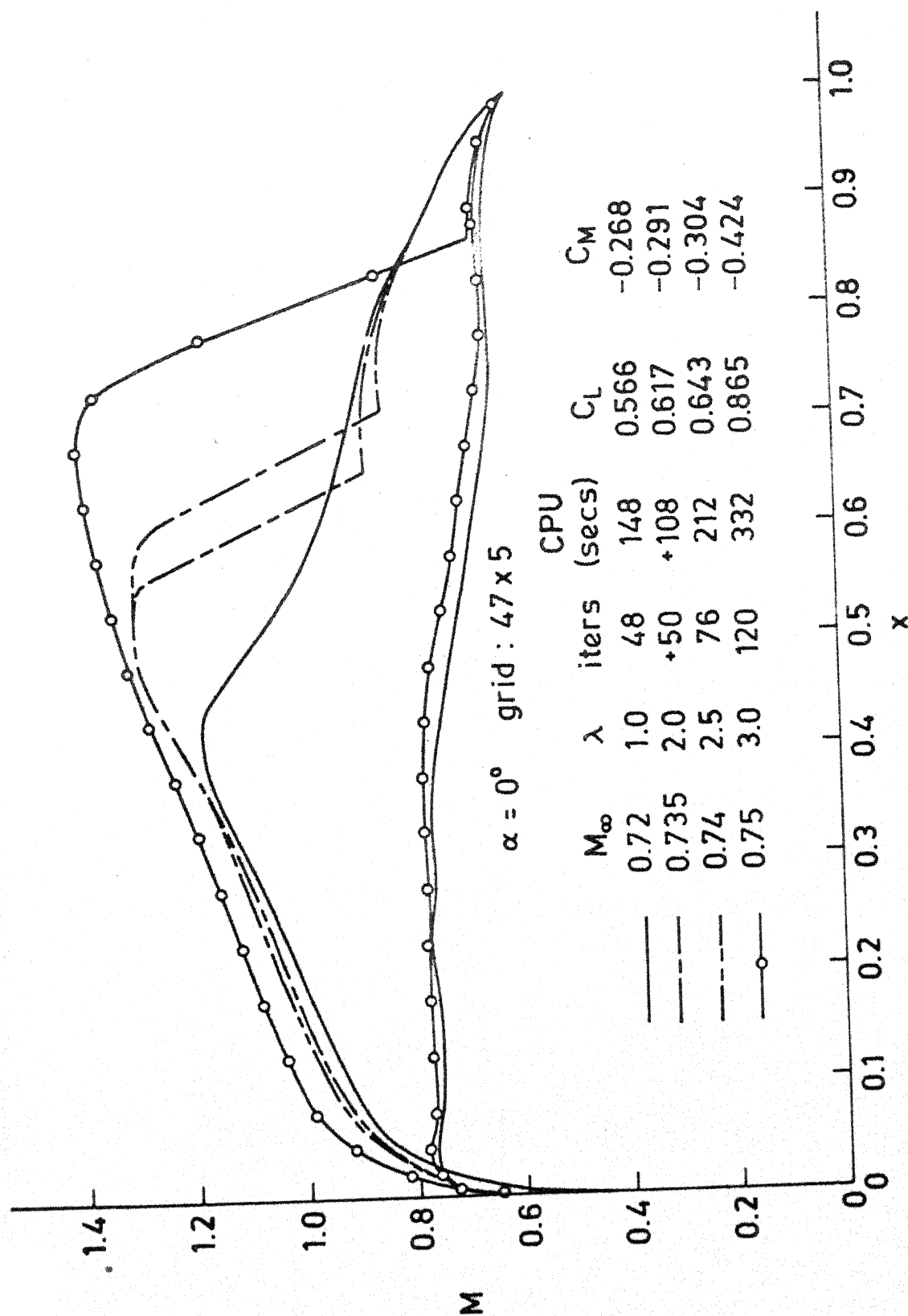
Fig. 3.24. Mach Number Distribution on the Surface of a NACA 64A410 Airfoil.

α = 0°   grid : 47 x 5

M_∞	λ	iters	CPU (secs)	C_L	C_M
0.72	1.0	48	148	0.566	-0.268
0.735	2.0	+50	+108	0.617	-0.291
0.74	2.5	76	212	0.643	-0.304
0.75	3.0	120	332	0.865	-0.424

Table 3.7

Co-ordinates for the NACA 64A410 Airfoil

Upper surface		Lower surface	
x	y	x	y
0	0	0	0
0.0035	0.00902	0.00650	-0.00678
0.00582	0.01112	0.00918	-0.00796
0.01059	0.01451	0.01441	-0.00969
0.02276	0.02095	0.02724	-0.01251
0.04749	0.03034	0.05251	-0.01592
0.07230	0.3765	0.07770	-0.01819
0.09737	0.04380	0.10263	-0.01996
0.14748	0.05366	0.15252	-0.02244
0.1977	0.06126	0.20230	-0.02406
0.248	0.06705	0.252	-0.02499
0.29834	0.07131	0.30166	-0.02537
0.34871	0.07414	0.35129	-0.02518
0.39910	0.07552	0.40090	-0.02436
0.44950	0.07552	0.45050	-0.02266
0.49989	0.07344	0.50011	-0.02024
0.55025	0.07040	0.54975	-0.01736
0.60057	0.06624	0.59943	-0.01418
0.65085	0.06106	0.64915	-0.01086
0.70108	0.05490	0.69892	-0.0076
0.75126	0.04780	0.74874	-0.0046
0.80151	0.03967	0.79849	-0.00229
0.85148	0.03018	0.84852	-0.00132
0.90104	0.02038	0.89896	-0.00076
0.95053	0.01028	0.94947	-0.00048
1.0	0.00021	1.0	-0.00021

Table 3.8

Surface Pressure Distribution for Super-critical Flow Past NACA 64A410 Airfoil

$M_\infty = 0.75$ $\alpha = 0°$ $\lambda = 3.5$ Grid: 47 x 5 $\alpha_1 = 0.5$

$C_L = 0.845$ $C_M = -0.413$ $C_D = -0.016$

x	C_p Upper	Lower
0	1.1206	1.1206
0.0005	1.0512	0.9088
0.0049	0.2581	0.2153
0.0193	-0.1467	0.0817
0.0461	-0.3947	0.0006
0.0811	-0.5712	0.005
0.125	-0.6791	-0.0114
0.175	-0.7712	-0.0266
0.225	-0.8489	-0.0386
0.275	-0.9185	-0.0477
0.325	-0.9910	-0.0534
0.375	-1.0738	-0.0588
0.425	-1.1486	-0.0511
0.475	-1.2032	-0.0159
0.525	-1.2443	0.0255
0.575	-1.2720	0.0675
0.625	-1.2880	0.1114
0.675	-1.2863	0.1553
0.725	-1.2352	0.1997
0.775	-0.9688	0.2419
0.83	-0.2799	0.2407
0.89	0.1887	0.2221
0.95	0.2275	0.2608
0.99	0.3384	0.3456
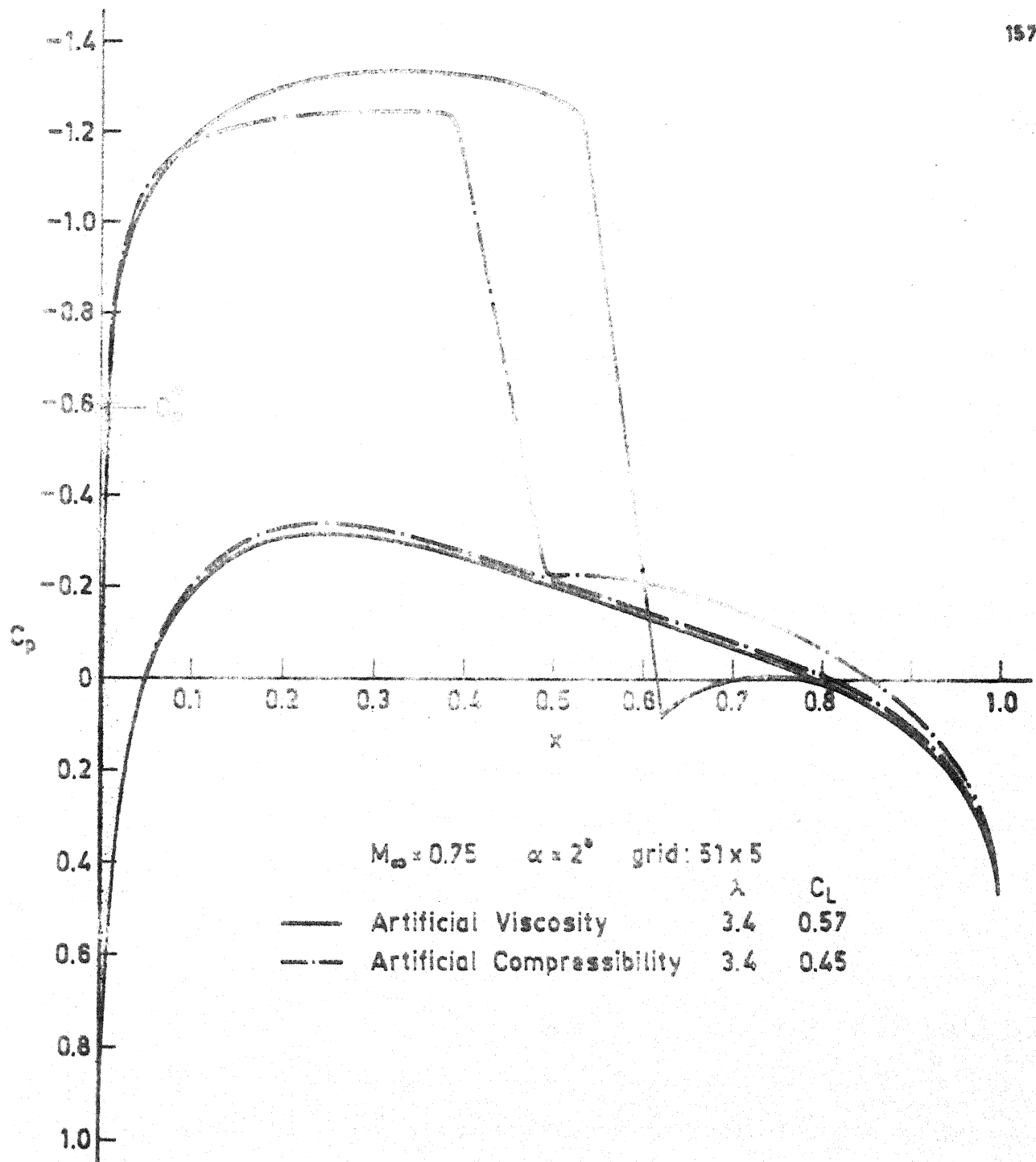
Fig. 3.25. Surface Pressure Distribution over a NACA 0012 Airfoil using Artificial Compressibility.

shock on a 256 x 64 grid and a smeared shock on a 64 x 16 grid at 0.5 chord. The present calculation on a very coarse mesh of 47 x 5 has practically smeared out the shock.

The last example of a super-critical calculation, Figure 3.25, shows a comparison of the results obtained for a NACA 0012 airfoil, $M_\infty = 0.75$, $\alpha = 2°$ using artificial compressibility and artificial viscosity. The use of artificial compressibility as given in Section 3.2.3 is seen to produce a surface pressure distribution ($C_L = 0.45$) which is closer to the non-conservative results ($C_L = 0.444$) than conservative results ($C_L = 0.58$).

The CPU time needed for the NACA 64A410 calculations was about 3 seconds per iteration. For the NACA 0012 profile, it was about 1.5 seconds per iteration. The computational effort however was the same for both cases. The difference was found to arise because of the use of virtual core for the NACA 64A410 calculations which caused swapping time to be added to the execution time, resulting in larger CPU time per iteration.

The maximum local Mach number reached in these examples is slightly more than 1.3 which is the upper limit of validity of the weak shock assumption. Getting the iteration scheme to converge for still stronger super-critical flows was found to be difficult. The iteration count increases rapidly as the flow becomes more and more super-critical. There are however reasons to believe that with the systematic use of artificial time-dependent damping terms, it may be possible to attain converged solutions for super-critical cases stronger than those presented here.

3.5.5. Iterative Development of the Solutions and Convergence
Performance of the Scheme

Figures 3.26 and 3.27 illustrate the convergence performance of the iterative scheme for symmetric flow past a NACA 0012 airfoil.

Figure 3.26 shows the iterative development of the surface Mach number distribution at $M_\infty = 0.84$. The value of λ was set at 4.0 for this case. Artificial time-dependent term $\alpha_1 u_t$ had to be used to achieve convergence in this case, with $\alpha_1 = 0.5$. For smaller values of α_1, the iteration scheme tended to diverge. For $\alpha_1 = 0.5$, convergence occurred in about 110 iterations. The figure shows the surface Mach number distribution plotted for every 20 iterations. It is seen that the shock location has stabilised itself around 0.70 chord in about 20 iterations and thereafter changes very little. At the end of about 40 iterations, an over expansion tendency is noticeable just ahead of the shock. This however does no damage to the further development of the solution which, to plotting accuracy, has reached its converged value in about 60 iterations. The rest of the iterations are carried out to achieve finer convergence based on the criterion on $\Delta \bar{g}$, see Section 3.4.4.

The plot of $\Delta \bar{g}$ for this case is also shown in Figure 3.26. It is seen that the convergence behaviour for this case is not uniform and that there are large fluctuations in the initial stages. These fluctuations die out in about 50 iterations after which, convergence though slow is fairly smooth.

Figure 3.27 shows the effect of the addition of an artificial time-dependent term $\alpha_1 u_t$ to the field source term $\bar{g}$
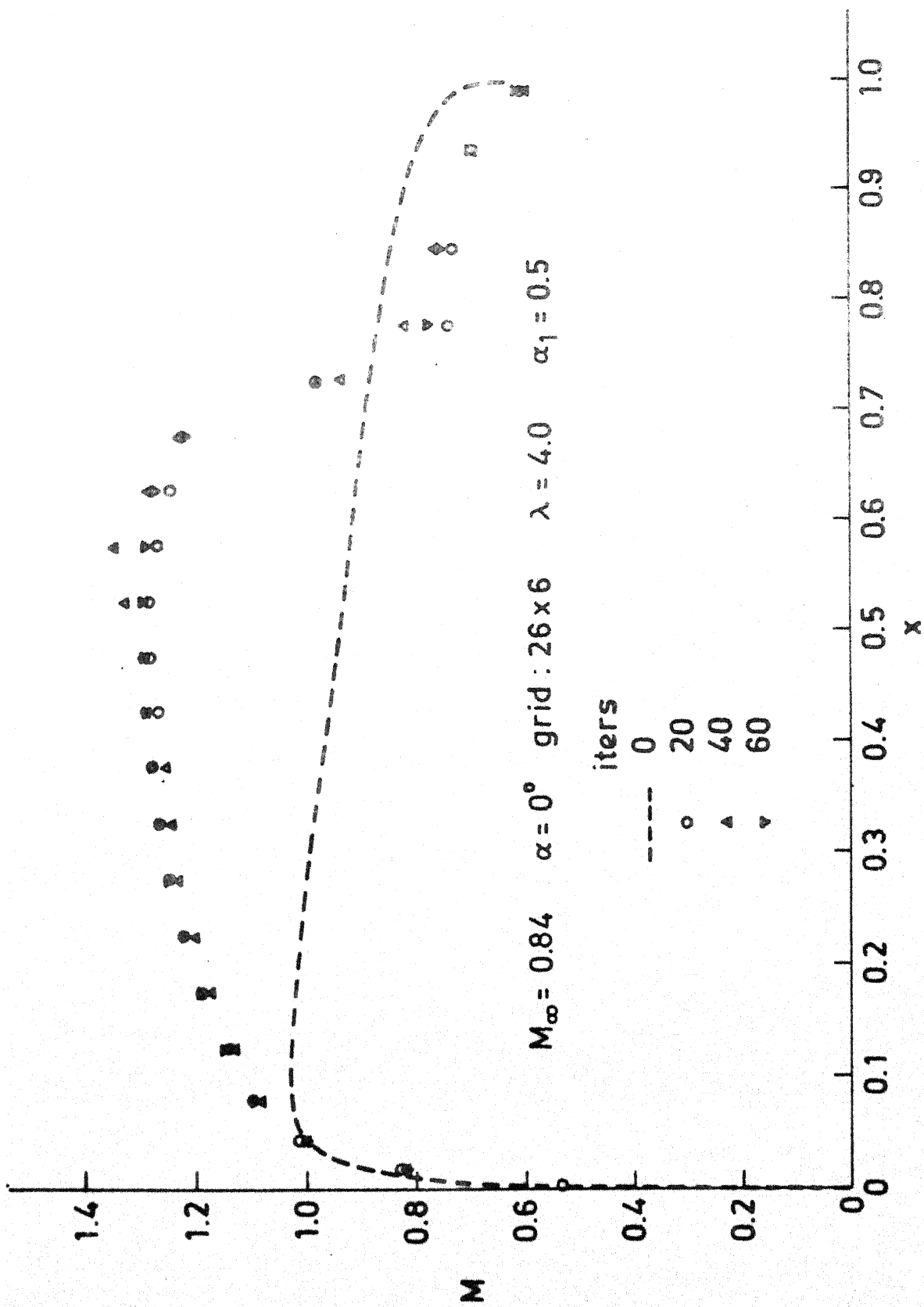
Fig. 3.26. Iterative Development : Mach Number Distribution on the Surface of a NACA 0012 Airfoil.
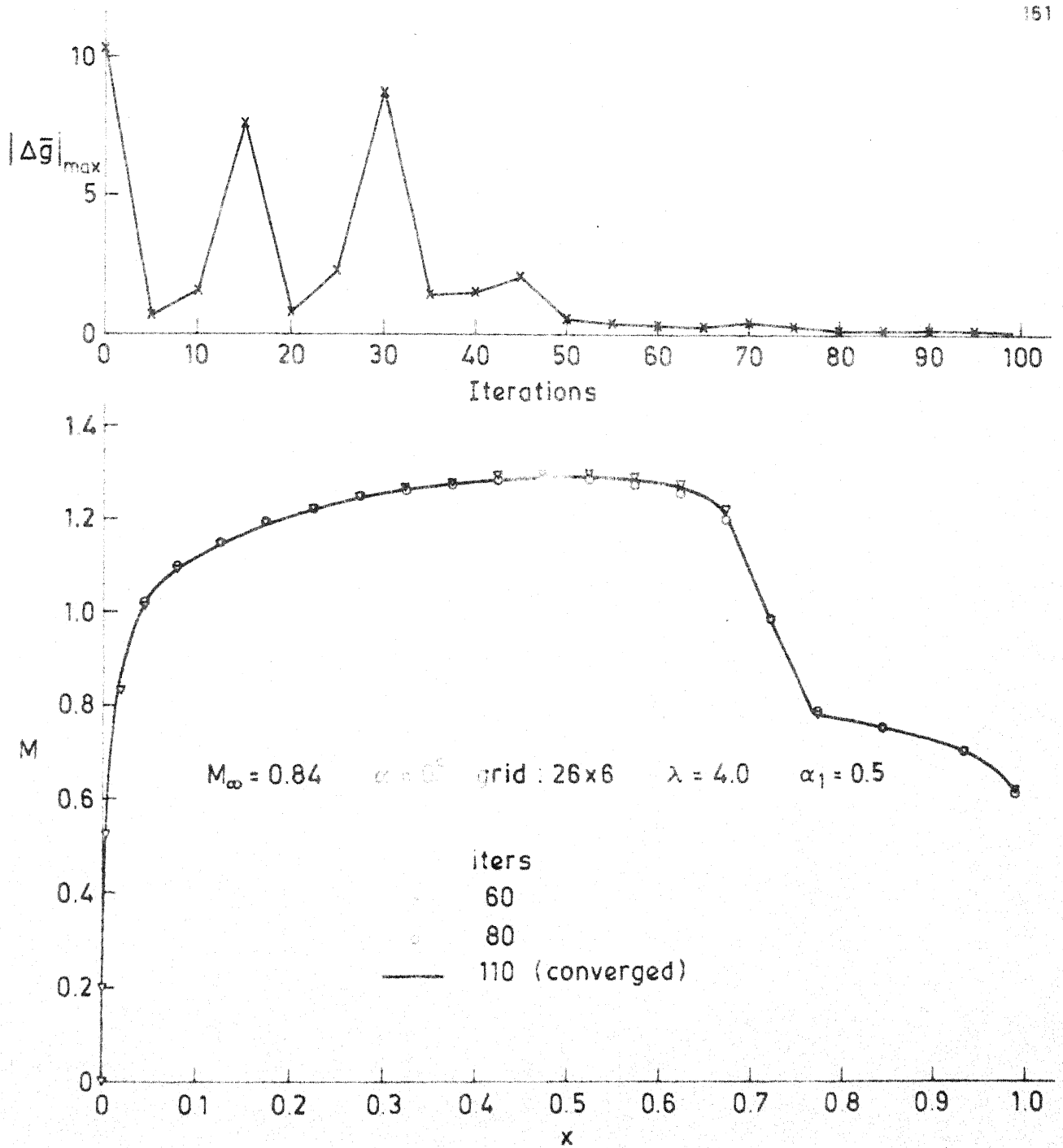
Fig. 3.26(contd). Iterative Development: Mach Number Distribution on the Surface of a NACA 0012 Airfoil.
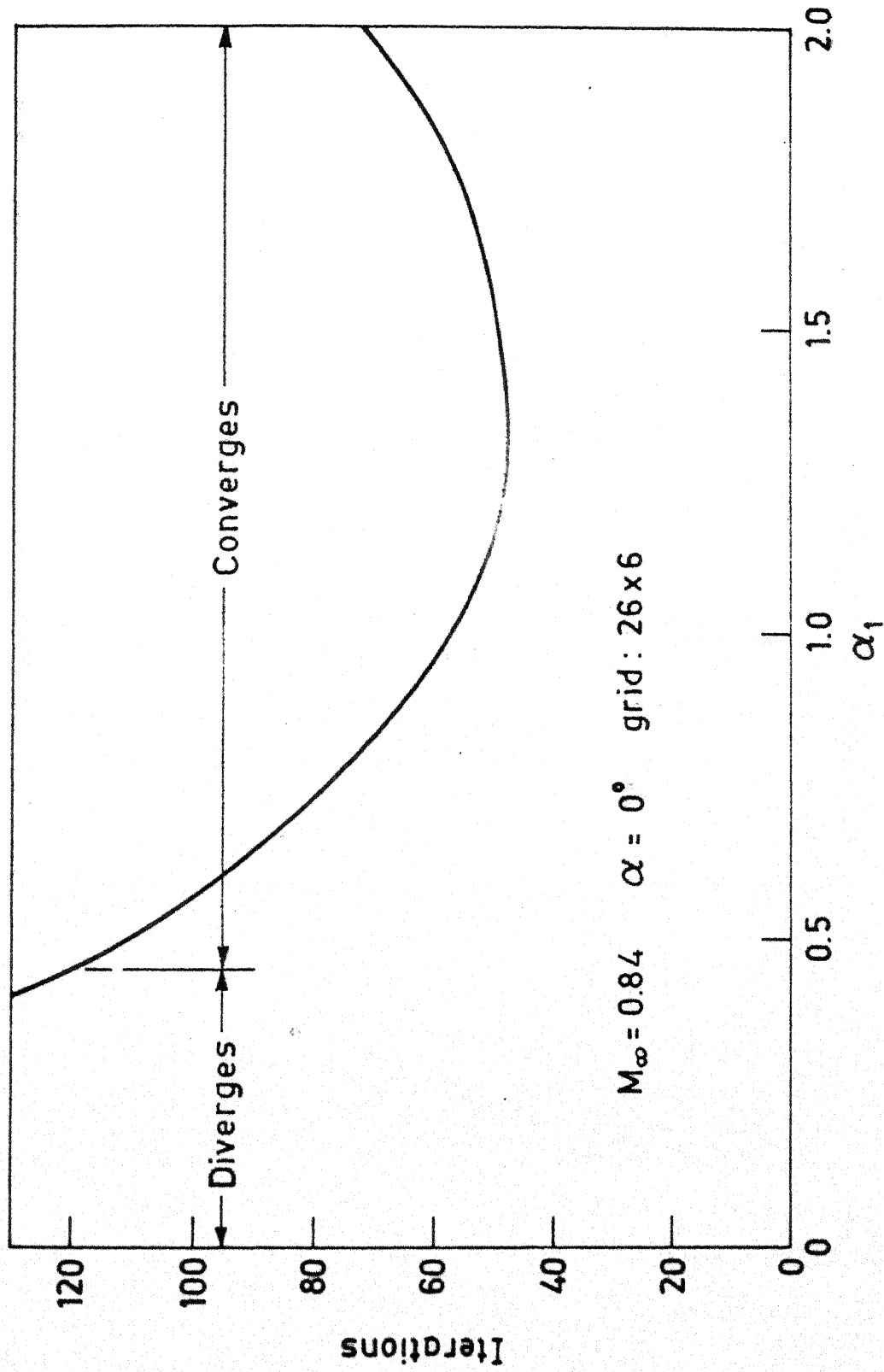
Fig. 3.27. Variation of Number of Iterations for Convergence with α_1 $(\alpha_2 = 0)$ for a NACA 0012 Airfoil.
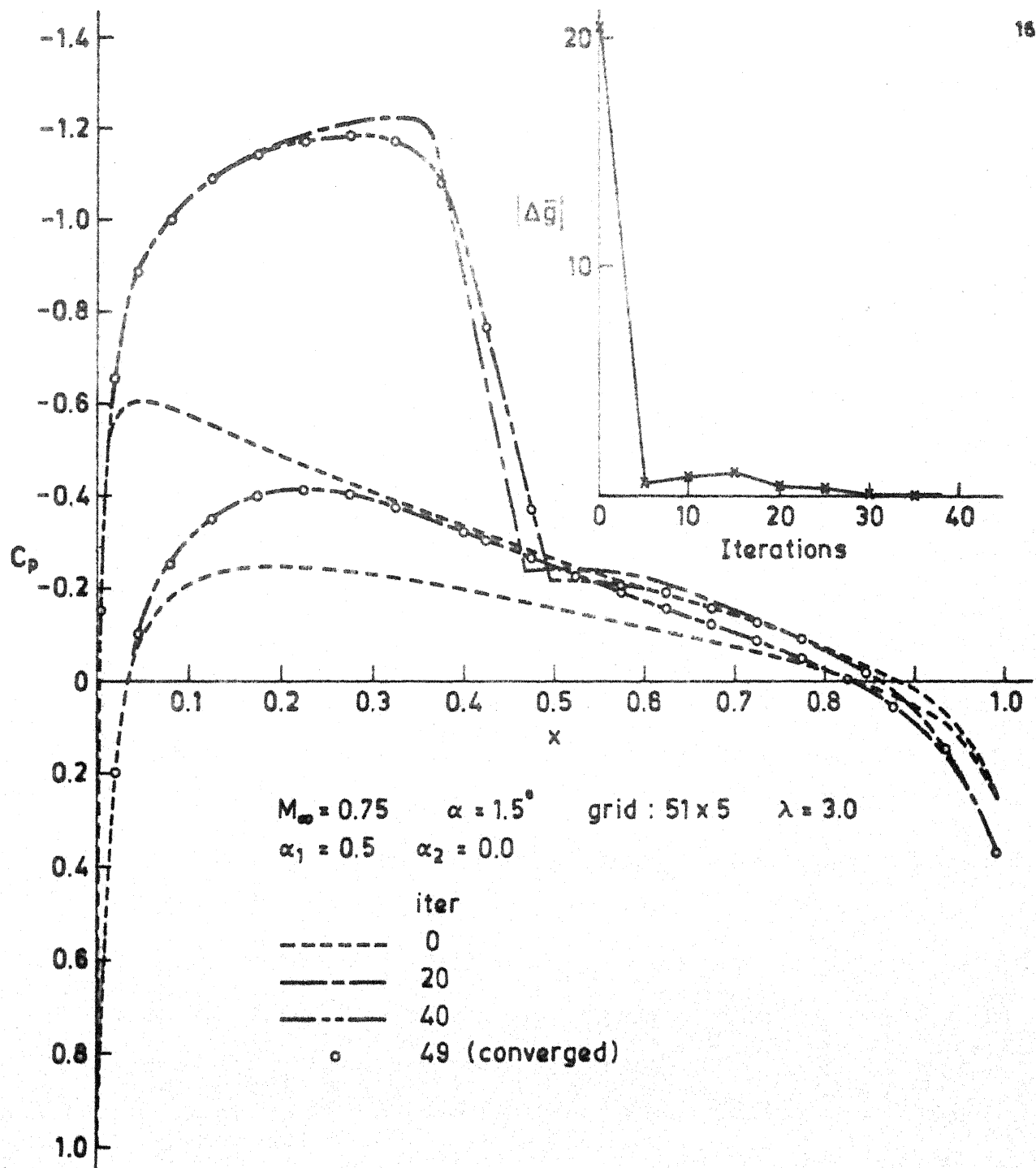
Fig. 3.28. Iterative Development : Surface Pressure Distribution on a NACA 0012 Airfoil.

during the iterative calculations. With $\alpha_1 = 0$, the iterative scheme diverges for this Mach number. Convergence occurs in about 110 iterations when $\alpha_1 = 0.5$ and as α_1 is increased to 1.25, convergence is achieved in just about 50 iterations. With further increase in α_1 however, the iteration count begins to increase.

Figure 3.28 shows the iterative development of the surface pressure distribution for a lifting flow, the airfoil again being NACA 0012. For this case, $M_\infty = 0.75$, $\alpha = 1.5°$ and $\alpha_1 = 0.5$. Here again we notice a tendency for peaky velocities to develop ahead of the shock at the end of 20 iterations. The shock position has not stabilised yet. The solution at the end of 40 iterations when $\Delta \bar{g} \leq 0.025$, is however indistinguishable from the final converged solution after 49 iterations. The lower surface pressure distribution has expectedly converged in less than 20 iterations.

The plot of $\Delta \bar{g}$ shows that the convergence rate has declined as the solution is approached for $\Delta \bar{g} < 0.5$. The convergence on the other hand for this case is more or less smooth. This is due to the moderate size of the supersonic region and the relatively weak shock.

3.5.6. Comparison of FP and TSP Solutions

A few comparisons of the FP and TSP solutions obtained for a 10 percent thick parabolic arc airfoil will be presented in this section. Both FP and TSP calculations were carried out without the use of artificial time-dependent terms.
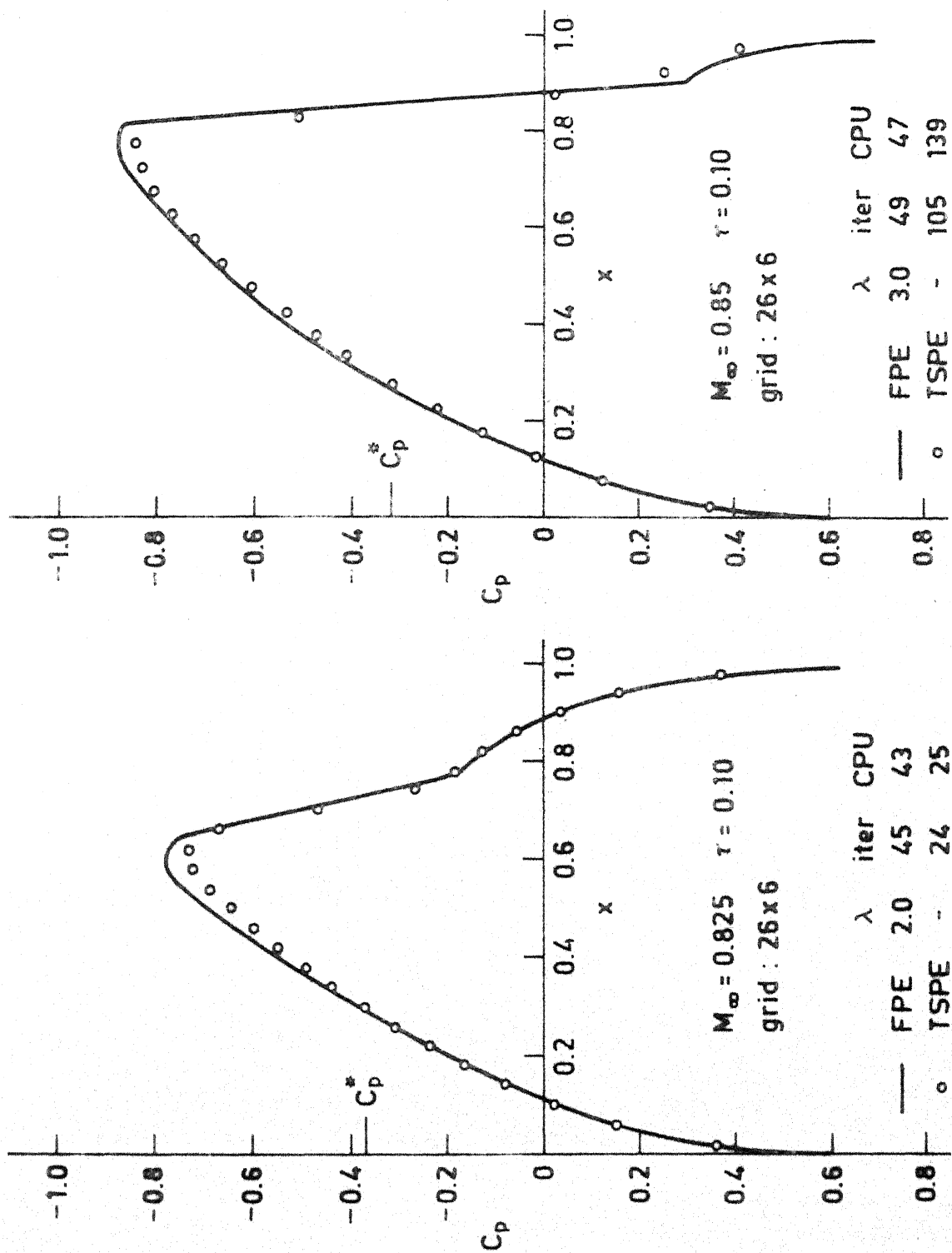
Fig. 3.29. Comparison of FP and TSP Solutions for a Parabolic Airfoil.
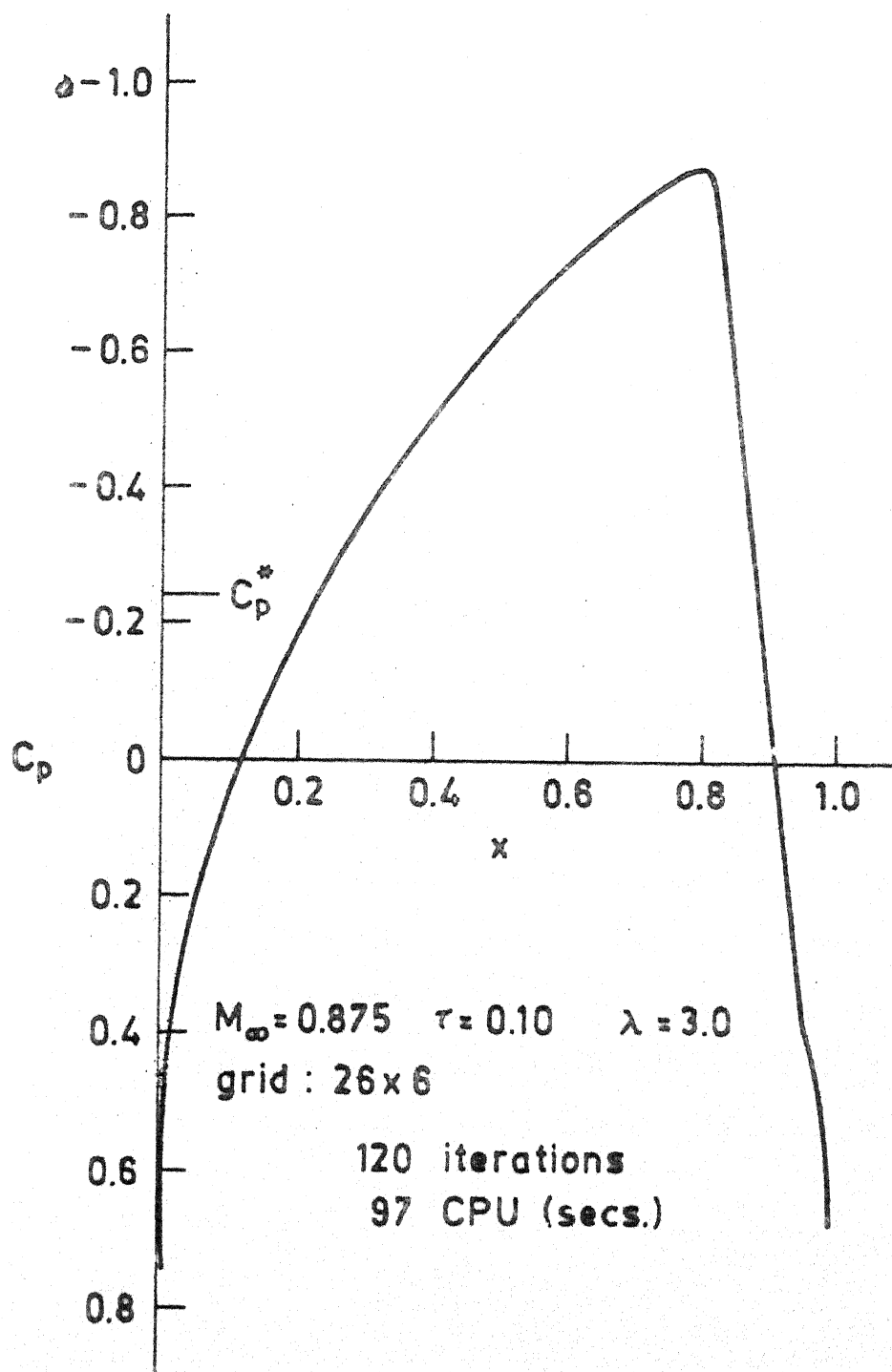
Fig. 3.30. Surface Pressure Distribution on a Parabolic Arc Airfoil.

Figures 3.29a and b shows the computed FP and TSP pressure distributions at two different free-stream Mach numbers, 0.825 and 0.85. The agreement between the two results is seen to be rather good. The FP shock is located slightly downstream of the TSP shock and is also somewhat stronger. Figure 3.30 shows the converged FP solution obtained for $M_\infty = 0.875$ whereas the TSP code failed to converge for this case, even with the inclusion of artificial time-dependent terms.

The computing times for the FP and TSP solutions for $M_\infty = 0.825$ appear to be comparable. However it must be kept in mind that the FP code, for symmetric cases, uses about 4 times the core required by a TSP code and hence is four times as costly. This advantage of the TSP code seems to be off set for $M_\infty = 0.85$, where it takes nearly 4 times as much CPU time as the FP code which converges rather quickly. This, as well as the fact that converged solution could be obtained for $M_\infty = 0.875$, suggest that the FP formulation, for the IEM, is numerically more stable than the TSP formulation.

CHAPTER IV

CONCLUSIONS

From the results and discussion presented in Chapters II and III, it is possible to draw the following conclusions as regards the use of the Integral Equation Method for the computation of axisymmetric and plane, two-dimensional super-critical flows with weak shock discontinuities.

In general, for both TSPE and FPE formulations the shock-capturing field panel technique of IEM seems to work satisfactorily for flows in which the embedded supersonic regions are large and terminated by relatively strong shocks. The captured shocks in such cases are sharp and agree fairly well with comparable results of FDM or FEM, in terms of both shock strength and location. For flows which are mildly super-critical, the field panel technique tends to smear the shock, and both the strength and location of the calculated shocks differ considerably from the FDM or FEM results. The situation here can perhaps be improved by resorting to a finer mesh but this has the undesirable consequence of increasing the core requirement.

The wall interference calculations for the axisymmetric (TSPE) case show that the IEM is also well suited for dealing with different boundary conditions, e.g. perforated wall, slotted wall, solid wall, free jet or specification of a known pressure or velocity distribution at the tunnel wall in the problem. These boundary conditions then appear as additional integrals in the integral equation for the axial velocity component.

Modifications to the program code may therefore be easily incorporated in both TSP and FP numerical algorithms, to take into account different wall conditions.

Convergence rates achieved for the present straight-forward algorithms may be considered satisfactory. Even for the strongest super-critical cases for which computations were carried out, convergence was achieved in about 120 iterations. Considering that a detailed numerical study of the convergence characteristics of the present scheme has not been carried out, it is reasonable to believe that improvements in convergence rates are possible if more attention is paid to this aspect.

For the TSP version of the axisymmetric problem, two iteration schemes — the quasi-Newton Scheme and the Direct Iteration Scheme — were used. Results of the present investigation suggest that for free air calculations, the QNS is faster and more stable. For the wall interference case, the DIS appears to be more stable than the QNS.

Unlike the semi-direct methods which use fast Poisson solvers and generally diverge for even slightly super-critical flows without the use of artificial time-dependent damping terms, the present integral equation approach, especially the FP version for plane flow, has shown that the inclusion of artificial viscosity is sufficient to achieve convergence for most cases arising in inviscid, isentropic flow. However, the inclusion of artificial time-dependent terms has been found to be beneficial for achieving stability and convergence for strong super-critical cases. This is true for both the TSP and FP formulations presented in the thesis.

A theoretical analysis of the effect of the presence of the time-dependent terms on iterative convergence has not been attempted in the present work. Therefore the range of values of the parameters ω, α_1 and α_2 within which the iteration scheme converges is not known in advance. This restricts the usefulness of these parameters, at present, to only obtaining convergence for strong super-critical cases which otherwise might diverge.

The viscosity parameter λ in FP formulation is found to be significant for its effect on the strength of the captured shock. The criterion proposed in the present work, by which a solution is chosen from a sequence of solutions obtained for decreasing values of λ, is to be considered tentative, since it has not been tested for a wide range of airfoils. The problem here, as already noted, is that the artificial viscosity term, being a completely arbitrary numerical artifice, there is no satisfactory consideration by which the amount of dissipation can be prescribed in advance. In the absence of suitable physical criteria, this term is manipulated only by numerical consideration such as stability or captured shock characteristics. Hence, small and at times significant, uncertainties are present on the computed surface pressure distributions. However it must be noted that this uncertainty is not a peculiarity of the IEM alone. It is common to other potential flow calculations also, see for example [36].

In the present approach to the FP formulation, the internal singularity model of Basu [104] for incompressible flow has been successfully adopted for the calculation of compressible

flows with or without shocks, by considering the non-linear
terms as an additional field source distribution. The advantage
of the internal singularity model as compared to the surface
singularity models, particularly for purposes of compressible
flow calculations with shocks, lies in the fact that the former
yields satisfactory incompressible solutions with fewer number
of unknowns. Clearly then, the singularity model used to
generate the incompressible solution to the Laplace equation
must be satisfactory if the final non-linear solution is to be
satisfactory. As mentioned in Chapter III, the present internal
singularity model fails to yield satisfactory results for
airfoils with nose radius of the order of 0.1 or greater.
Further the calculated solution in such cases are sensitive to
the choice of the leading edge gap δ. For these reasons it
is clear that there is a need to either modify the present
model or search for alternative surface singularity models
which retain the advantage of the internal singularity model of
requiring fewer unknowns, see for example [110].

Lastly, it must be recognized that, IEM algorithms
generally need, for a given problem and mesh size, several times
more core than a corresponding finite-difference algorithm. The
difference in the core requirements arises largely due to the
need to store the influence co-efficients. The TSP and the FP
versions of the IEM algorithms, respectively need for a given
grid, 2 to 3 times and 4 to 6 times the core required by the
corresponding finite-difference algorithms. Some core economy
can be achieved for the case of TSP algorithms as compared to
FP algorithms because the former calculations are carried out

on rectangular grids which give the influence co-efficients
certain symmetry properties. Even so, for a given problem and
mesh size, the FDM may seem attractive purely from the point of
view of core economy. However some of the features of the IEM
enable an overall reduction in the total number of grid points
to be achieved, which reduces the core requirement considerably.
Firstly, as was evident from the discussion of both TSP and FP
results, the IEM gives satisfactory results with a much smaller
grid extent for the infinite domain problem than the finite-
difference and finite-volume methods. Secondly, to realise the
same accuracy, the IEM computations can be carried out on a
rather coarse mesh and therefore need fewer grid points than
the FEMs. For these reasons, the number of grid points used in
IEM are generally 1/2 or 1/4 the number used in equivalent
finite-difference or finite-volume methods.

REFERENCES

1. Nieuwland, G.Y. and Spee, B.M., "Transonic Airfoils: Recent Developments in Theory, Experiment, and Design", Annual Review of Fluid Mechanics, Vol. 5, 1973.

2. Holder, D.W., "Transonic Flow Past Two-Dimensional Airfoils", J. of Roy. Aero. Soc., Vol. 68, pp. 501-516, 1964.

3. Chapman, D.R., "Computational Aerodynamics Development and Outlook", AIAA Journal, Vol. 17, pp. 1293-1313, 1979.

4. Tai, T.C., "Transonic Inviscid Flow Over Lifting Airfoils by the Method of Integral Relations", AIAA Journal, Vol. 12, pp. 798-804, 1974.

5. Murman, E.M. and Cole, J.D., "Calculation of Plane Steady Transonic Flows", AIAA Journal, Vol. 9, pp. 114-121, 1971.

6. Murman, E.M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods", AIAA Journal, Vol. 12, pp. 626-633, 1974.

7. Lax, P.D. and Wendroff, B., "Systems of Conservation Laws", Comm. Pure Appl. Math., Vol. 13, pp. 217-237, 1960.

8. Jameson, A., "Iterative Solutions of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1", Comm. Pure Appl. Math., Vol. 27, pp. 283-309, 1974.

9. Albone, C.M., "A Finite-Difference Scheme for Computing Super-Critical Flows in Arbitrary Co-ordinate Systems", ARC CP No.1314, July 1974.

10. Hafez, M.M. and Cheng, H.K., "Convergence Acceleration and Shock Fitting for Transonic Flow Computations", AIAA Journal, Vol. 15, pp. 329-336, 1977.

11. Jameson, A., "Accelerated Iteration Schemes for Transonic Flow Calculations Using Fast Poisson Solvers", New York University ERDA Report COO-3077-82, March 1975.

12. Ballhaus, W.F., Jameson, A. and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems", AIAA Journal, Vol. 16, pp. 573-579, 1978.

13. Holst, T.L. and Ballhaus, W.F., "Fast, Conservative Schemes for the Full-Potential Equation Applied to Transonic Flows", AIAA Journal, Vol. 17, pp. 145-152, 1979.

14. Ballhaus, W.F. and Steger, J.L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equation", NASA TM X-73082, 1975.

15. Holst, T.L., "An Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh", AIAA Journal, Vol. 17, pp. 1038-1045, 1979.

16. Holst, T.L., "Fast Conservative Algorithm for Solving the Transonic Full-Potential Equation", AIAA Journal, Vol. 18, pp. 1431-1439, 1980.

17. Federenko, R.P., "A Relaxation Method for Solving Elliptic Difference Equations", USSR Comput. Math. Math. Phys., pp. 1092-1096, 1962.

18. Federenko, R.P., "The Speed of Convergence of One Iterative Process", USSR Comput. Math. Math. Phys., Vol. 4, pp. 227-235, 1964.

19. Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems", Math. of Computation, Vol. 31, pp. 333-390, 1977.

20. South, J.C. and Brandt, A., "Application of a Multi-Level Grid Method to Transonic Flow Calculations" in Transonic Flow Problems in Turbomachinery, Ed. Adamson, T.C. and Platzer, M.F., Hemisphere, Washington, 1977.

21. Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple-Grid Method", AIAA Paper No. 79-1458, Proc. of AIAA CFD Conf., July 1979, Williamsburg, pp. 122-146.

22. McCarthy, D.R. and Reynher, T.A., "A Multi-Grid Code for Three-Dimensional Transonic Potential Flow About Axisymmetric Inlets at Angle of Attack", AIAA Paper 80-1365, July 1980.

23. Martin, E.D. and Lomax, H., "Rapid Finite-Difference Computation of Subsonic and Transonic Aerodynamic Flows", AIAA Journal, Vol. 13, pp. 579-586, 1975.

24. Martin, E.D., "A Fast Semi-Direct Method for Computing Transonic Aerodynamic Flows", Proc. 2nd AIAA CFD Conf., Hartford, pp. 162-174, June 1975.

25. Jameson, A., "Transonic Flow Calculation", in 'Numerical Methods in Fluid Dynamics', Ed. H.J. Wirz and J.J. Smolderen, Hemisphere Publishing Corporation, 1978.

26. Argyris, Y.H. and Mareczek, G., "Potential Flow Analysis by Finite-Elements", Ingenier-Archiv, Jan. 1973.

27. Periaux, J., "Three Dimensional Analysis of Compressible Potential Flows with the Finite-Element Method", Int. J. Num. Meth. Engg., 9, 775-831, 1975.

28 Hafez, M.M., Wellford, L.C. and Murman, E.M., "Finite-Elements and Finite Differences for Transonic Flow Calculations", Finite Elements in Fluids, Vol. 3, Ed. Gallagher et al, John Wiley, 1978.

29. Wellford, L.C. and Hafez, M.M., "An Implicit Velocity Formulation for the Small Disturbance Transonic Flow Problem Using Finite-Elements", Symp. Applications of Computer Methods in Engg., University of Southern California, Aug. 1977.

30. Phares, W.J. and Kneile, K.R., "Solutions to the Eulerian Equations by the Finite-Element Method with an Application to Transonic Flows", Arnold Engg. Dev. Center (ARO Inc.), AEDC-TR-76-86 (1976).

31. Hafez, M.M., Wellford, L.C., Merkle, C.L. and Murman, E.M., "Numerical Computation of Transonic Flows by Finite-Element and Finite-Difference Methods", Flow Research Report No. 70, Jan. 1977.

32. Chan, S.T.K. and Brashears, M.R., "Finite-Element Analysis of Transonic Flow", AFFDL-TR-74-11, March 1974.

33. Wellford, L.C. and Hafez, M.M., "Mixed Finite-Element Models and Dual Iterative Methods for Transonic Flow", 2nd Inter. Symp. Finite-Element Methods in Flow Problems, St. Margherita Ligure, Italy, June 1976.

34. Glowinski, R. and Pironneau, O., "On the Computation of Transonic Flows", France-Japanese Conf. on Funct. Anal. and Num. Anal., Tokyo, Kyoto, Sept. 1976.

35. Eberle, A., "Transonic Flow Computations by Finite-Elements: Airfoil Optimisation and Analysis", Recent Dev. in Theor. & Exp. Fluid Mech., Ed. U. Muller, K.G. Roesner and B. Schmidt, Springer-Verlag, 1979.

36. Habashi, W.G. and Hafez, M.M., "Finite-Element Solutions of Transonic Flow Problems", AIAA Journal, Vol. 20, pp. 1368-1376, 1982.

37. Bristeau, M.O., "Application of a Finite Element Method to Transonic Flow Problems Using an Optimal Control Approach", Computational Fluid Dynamics, Ed. W. Kollmann, Hemisphere Publishing Corporation, 1980.

38. MacCormack, R.W. and Paullay, A.J., "The Influence of the Computational Mesh on Accuracy for Initial Value Problems with Discontinuities or Non-unique Solutions", Computers and Fluids, Vol. 2, pp. 339-361, 1974.

39. Rizzi, A., "Transonic Solutions of the Euler Equations by the Finite-Volume Method", Symposium Transsonicum II, Goettingen, pp. 567-574, Sept. 1975.

40. Caughey, D.A. and Jameson, A., "Numerical Calculation of Transonic Potential Flow About Wing-Body Combinations", AIAA Journal, Vol. 17, pp. 175-181, 1979.

41. Jameson, A. and Caughey, D.A., "A Finite-Volume Method for Transonic Potential Flow Calculations", Proc. of AIAA 3rd CFD Conf., Hartford, Conn., pp. 35-54, June 1977.

42. Caughey, D.A. and Jameson, A., "Progress in Finite-Volume Calculations for Wing-Fuselage Combinations", AIAA Paper 79-1513, also, AIAA Journal, Vol. 18, pp. 1281-1288, 1980.

43. Oswatitsch, K., "Die Geswindigkeitsverteilung bie Lokalen Uberschallgebeiten an flachen Profilen", ZAMM, Vol. 30, pp. 17-24, 1980.

44. Gullstrand, T.R., "The Flow Over Symmetrical Airfoils Without Incidence in the Lower Transonic Range", KTH Aero TN 20, 1951.

45. Gullstrand,T.R., "The Flow Over Symmetrical Airfoils Without Incidence at Transonic Speeds", KTH Aero TN 24, 1952.

46. Gullstrand, T.R., "A Theoretical Discussion of Some Properties of Transonic Flow Over Two-Dimensional Symmetrical Airfoils at Zero Lift with a Simple Method to Estimate the Flow Properties", KTH Aero TN 25, 1952.

47. Gullstrand, T.R., "The Flow Over Two-Dimensional Airfoils at Incidence in the Transonic Speed Range", KTH Aero TN 27, 1952.

48. Gullstrand, T.R., "Transonic Flow Past Two-Dimensional Airfoils", Zeitschrift fur Flugwissen Schaften, Vol. 1, pp. 38-46, 1953.

49. Spreiter, J.R. and Alksne, A.Y., "Theoretical Prediction of Pressure Distributions on Non-lifting Airfoils at High Subsonic Speed", NACA Report 1217, 1955.

50. Oswatitsch, K. and Keune, F., "The Flow Around Bodies of Revolution at Mach Number 1", Proc. Conf. on High Speed Aeronautics, Polytech. Inst. of Brooklyn, pp. 113-131, 1954. 1955.

51. Heaslet, M.A. and Spreiter, J.R., "Three-Dimensional Transonic Flow Theory Applied to Slender Wings and Bodies", NACA Tech. Rep. 1318, 1956.

52. Klunker, E.B., "Contribution to Methods for Calculating the Flow About Thin Lifting Wings at Transonic Speeds - Analytic Expressions for the Far Field", NASA TN D-6530, 1971.

53. Nørstrud, H., "Numerische Losungen fur Schallnahe Stromungen und Ebene Profile", Zeitscrift fur Flugwissenschaften, Heft 5, Vol. 18, pp. 149-157, 1970.

54. Nørstrud, H., "The Transonic Airfoil Problem with Embedded Shocks", Aero. Quart., Vol. XXIV, Part 2, pp. 129-138, 1973.

55. Nørstrud, H., "Transonic Flow Past Lifting Wings", AIAA Journal, Vol. 11, No. 5, pp. 754-757, 1973.

56. Nørstrud, H., "High Speed Flow Past Wings", NASA CR-2246, 1973.

57. Nixon, D. and Hancock, G.J., "High Subsonic Flow Past a Steady Two-Dimensional Aerofoil", ARC CP 1280, 1974.

58. Nixon, D., "An Aerofoil Oscillating at Low Frequency in a High Subsonic Flow", ARC CP 1285, 1974.

59. Niyogi, P., "Recent Developments in Integral Equation Method in Transonic Flow", Proc. Indian Acad. Sci. (Engg. Sci.), Vol. 3, Pt. 2, pp. 143-167, 1980.

60. Niyogi, P., "Transonic Flow Past Thin Wings", Proc. Indian Acad. Sci. (Engg. Sci.), Vol. 4, Pt. 3, pp. 347-361, 1981.

61. Kraft, E.M. and Lo, C.F., "Analytical Determination of Blockage Effects in a Perforated-Wall Transonic Wind Tunnel", AIAA Journal, Vol. 15, pp. 511-517, 1977.

62. Mokry, M., "Integral Equation Method for Subsonic Flow Past Airfoils in Ventilated Wind Tunnels", AIAA Journal, Vol. 13, pp. 47-53, 1975.

63. Nixon, D., "Extended Integral Equation Method for Transonic Flows", AIAA Journal, Vol. 13, pp. 934-936, 1975.

64. Nixon, D., "A Comparison of Two-Integral Equation Methods for High Subsonic Lifting Flows", Aero. Quart., Vol. XXVI, Part 1, pp. 56-58, 1975.

65. Nixon, D. and Patel, J., "The Evaluation of an Integral Equation Method for Two-Dimensional Shock Free Flows", Aero. Quart., Vol. XXVI, Part 1, pp. 59-70, 1975.

66. Nixon, D. and Hancock, G.J., "Integral Equation Methods a Reappraisal", Symposium Transonicum II, 1975.

67. Nixon, D., "Calculation of Transonic Flows Using an Extended Integral Equation Method", AIAA Journal, Vol. 15, pp. 295-296, 1977.

68. Ogana, W. and Spreiter, J.R., "Derivation of an Integral Equation for Transonic Flows", AIAA Journal, Vol. 15, pp. 281-283, 1977.

69. Ogana, W., "Numerical Solutions for Subcritical Flow by a Transonic Integral Equation Method", AIAA Journal, Vol. 15, pp. 444-446, 1977.

70. Arora, N.L. and Agarwal, J.P., "Axisymmetric Transonic Flow Past Slender Bodies by an Integral Equation Method", AIAA Journal, Vol. 18, pp. 606-608, 1980.

71. Arora, N.L., "Transonic Flow Past Lifting Airfoils Including Wind Tunnel Wall Interference Effects", Proc. First Asian Congress of Fluid Mechanics, Dec. 8-13, 1980.

72. Lobo, M. and Sachdev, P.L., "Some Direct Numerical Approaches to the Solution of the Singular Non-linear Transonic Integral Equation", Proc. Roy. Soc. London A 374, pp. 35-64, 1981.

73. Piers, W.J. and Slooff, J.W., "Calculation of Transonic Flow by Means of a Shock-Capturing Field Panel Method", AIAA Paper No. 79-1459, 1979.

74. Sachdev, P.L. and Lobo, M., "Application of Artificial Viscosity in Establishing Supercritical Solutions to the Transonic Integral Equation", Proc. Roy. Soc. London A 380, pp. 77-97, 1982.

75. Ravichandran, K.S., Arora, N.L. and Singh, R., "Axisymmetric Transonic Flow Past Slender Bodies Including Perforated Wall Interference Effects", AIAA Journal, Vol. 20, pp. 1557-1564, 1982.

76. Krupp, J.A., "The Numerical Calculation of Plane Steady Transonic Flow Past Thin Lifting Airfoils", BSRL Report D 180-12958-1, June 1971.

77. Ballhaus, W.F., "Some Recent Progress in Transonic Flow Computations", in 'Numerical Methods in Fluid Dynamics', eds. W.J. Wirz and J.J. Smolderen, pp. 155-235, Hemisphere publishing Corporation, 1978.

78. Hinson, B.L. and Burdges, K.P., "Evaluation of Three-Dimensional Transonic Codes Using New Correlation-Tailored Test Data", J. Aircraft, Vol. 18, pp. 855-861, 1981.

79. Luu, T.S., Goulmy, G. and Dulieu, A., "Calcul de l'ecoulement transsonique autour d'un profil en admettant la loi de compressibilite exacte", Association Technique Maritime et Aeronautique Session, 1972.

80. Stricker, R., "On an Integral Equation Method for Calculation of the Fully Non-linear Potential Flow About Arbitrary Section Shapes", Paper presented at the Euromech Colloquium 75 on 'The Calculation of Flow Fields by Means of Panel Methods', Rhode/Braunschweig, May 1976.

81. Golberg, M.A., "A Survey of Numerical Methods for Integral Equations" p. 53, 'Solution Methods for Integral Equations, Theory and Applications', Plenum Press, New York, 1979.

82. Sears, W.R., "Self-Correcting Wind Tunnels", Aeronautical Journal, Vol. 78, pp. 80-89, 1974.

83. Kemp, W.B. Jr., "Toward the Correctable-Interference Transonic Wind Tunnel", AIAA 9th Aerodynamic Testing Conf., June 1976, pp. 31-38.

84. Bailey, F.R., "Numerical Calculation of Transonic Flow About Slender Bodies of Revolution", TN-D-6882, 1971, NASA.

85. Sedin, Y.C.-J., "Axisymmetric Sonic Flow Computed by a Numerical Method Applied to Slender Bodies", AIAA Journal, Vol. 13, pp. 504-511, 1975.

86. Murman, E.M., "Computation of Wall Effects in Ventilated Transonic Wind Tunnels", AIAA Paper 72-1007, AIAA 7th Aerodynamics Testing Conf., Sept. 1972.

87. Kacprzynski, J.J., "Transonic Flow Field Past 2-D Airfoils Between Porous Wind Tunnel Walls with Non-linear Characteristics", AIAA Paper 75-81, AIAA 13th Aerospace Science Meeting, Jan. 1975.

88. Karlsson, K.R. and Sedin, Y.C.-J., "Numerical Design and Analysis of Optimal Slot Shapes for Transonic Test Sections — Axisymmetric Flows", J. Aircraft, Vol. 18, pp. 168-175, 1981.

89. Stahara, S.S. and Spreiter, J.R., "A Transonic Wind Tunnel Interference Assessment — Axisymmetric Flows", AIAA Journal, Vol. 18, pp. 63-68, 1980.

90. Jacocks, J.L., "An Investigation of the Aerodynamic Characteristic of Ventilated Test Section Walls for Transonic Wind Tunnels", Ph.D. dissertation, Univ. of Tennessee, Knoxville, Dec. 1976.

91. Baldwin, B.S., Turner, J. and Knechtel, E., "Wall Interference in Wind Tunnels with Slotted and Porous Boundaries at Subsonic Speeds", NACA TN 3176, 1954.

92. Ashley, H. and Landahl, M., "Aerodynamics of Wings and Bodies", Addison-Wesley Publishing Co., Reading, Mass., 1965.

93.   Richtmeyer, R.D., "Principles of Advanced Mathematical
        Physics", Springer-Verlag, New York, 1978.

94.   Grtega, J.M. and Rheinboldt, W.C., "Iterative Solutions
        of Non-linear Equations in Several Variables", Academic
        Press, New York, 1970.

95.   McDevitt, J.B. and Taylor, R.A., "Pressure Distributions
        at Transonic Speeds for Parabolic-Arc Bodies of Resolution
        Having Fineness Ratios of 10, 12 and 14", TN 4234,
        March 1958, NACA.

96.   McDevitt, J.B. and Taylor, R.A., "Pressure Distributions
        at Transonic Speeds for Slender Bodies Having Various
        Axial Locations of Maximum Diameter", TN 4280, July 1958,
        NACA.

97.   Krupp, J.A. and Murman, E.M., "Computation of Transonic
        Flows Past Lifting Airfoils and Slender Bodies", AIAA
        Journal, Vol. 10, pp. 880-886, 1972.

98.   Drougge, G., "An Experimental Investigation of the
        Interference Between Bodies of Revolution at Transonic
        Speeds, with Special Reference to the Sonic and Supersonic
        Area Rules", FFA. Rept. 83, 1959, Aeronautical Research
        Inst., Sweden.

99.   Bauer, F., Garabadian, P., Korn, D. and Jameson, A.,
        "Super-critical Wing Sections II", Lect. Notes in Economics
        and Mathematical Systems, Vol. 108, Springer-Verlag, 1975.

100.   Harten, A., "The Artificial Compression Method for Compu-
        tation of Shocks and Discontinuities", ICASE Rept. No.
        77-2, Feb. 1977.

101.   Hafez, M., South, J. and Murman, E., "Artificial Compre-
        ssibility Methods for Numerical Solutions of Transonic
        Full-Potential Equation", AIAA Journal, Vol. 17, pp. 838-
        844, 1979.

102.   Mercer, J.E., Geller, E.W., Johnson, M.L. and Jameson, A.,
        "Transonic Flow Calculations for a Wing in a Wind Tunnel",
        J. of Aircraft, Vol. 18, pp. 707-711, 1981.

103.   Hess, J.L. and Smith, A.M.O., "Calculation of Non-lifting
        Potential Flow About Arbitrary Two-Dimensional Bodies",
        Journal of Ship Research, Vol. 8, No. 2, pp. 22-44, Sept.
        1964.

104.   Basu, B.C., "A Mean Camber Line Method for Two-Dimensional
        Steady and Oscillatory Airfoils and Control Surfaces in
        Inviscid, Incompressible Flow", ARC CP No. 1391, Oct.
        1976.

105. Lock, R.C., "Test Cases for Numerical Methods in Transonic Flows", AGARD Rept. No. 575, 1970.

106. Jameson, A., "Transonic Flow Calculations", AIAA Professional Study Series, 1979.

107. Byrd, P.F. and Friedman, M.D., "Handbook of Elliptic Integrals for Engineers and Scientists", Springer-Verlag, 1971.

108. "Handbook of Mathematical Functions", Eds. M. Abramovitz and I.A. Stegun, Dover Publications, New York, pp. 591-592.

109. Abbott, I.A. and Von Doenhoff, A.E., "Theory of Wing Sections", Dover Publications, New York.

110. Maskew, B. and Woodward, F.A., "Symmetrical Singularity Model for Lifting Potential Flow Analysis", J. Aircraft, Vol. 13, pp. 733-734, 1976.

111. Steinhoff, J. and Jameson, A., "Multiple Solutions of the Transonic Potential Flow Equation", AIAA Journal, Vol. 20, pp. 1521-1525, 1982.

112. Sells, C.C.L., "Plane Sub-critical Flow Past a Lifting Airfoil", R.A.E. Tech. Rep. 67146, 1967.

APPENDIX A

EVALUATION OF EXPRESSIONS OCCURRING IN SECTION **2.3**

(i) Simplification of the Expression for I_{sw}

From Section 2.3, subsection (iiib) we have

$$I_{sw} = - r_w \iint_{S_w} (\emptyset \, \Psi_{r'} + \frac{p}{\beta} \, \Psi \, \emptyset_{x'})_{r'=r_w} \, d\theta' \, dx' \qquad (A-1)$$

$$= - \frac{r_w p}{\beta} \int_{-\infty}^{\infty} \int_0^{2\pi} \emptyset_{x'} \, \Psi \big|_{r'=r_w} \, d\theta' \, dx' -$$

$$r_w \int_{-\infty}^{\infty} \int_0^{2\pi} \emptyset \, \Psi_{r'} \big|_{r'=r_w} \, d\theta' \, dx'$$

Now consider each integral separately over θ' with $\Psi = \frac{1}{4\pi R}$ and $R = [(x - x')^2 + r^2 + r_w^2 - 2rr_w \cos\theta']^{1/2}$. Thus we obtain,

$$\int_0^{2\pi} \Psi \, d\theta' = \frac{1}{2\pi} \int_0^{\pi} \frac{d\theta'}{R} = \frac{1}{\pi} \frac{F(\pi/2, k)}{[(x - x')^2 + (r + r_w)^2]^{1/2}}$$

$$(A-2)$$

where $\quad k^2 = \dfrac{4rr_w}{(x - x')^2 + (r + r_w)^2}$

(see 291.00 of ref. 107).

Again

$$\int_0^{2\pi} \Psi_{r'} \, d\theta' = - \frac{1}{4\pi} \int_0^{2\pi} \frac{r_w - r \cos\theta'}{R^3} \, d\theta'$$

$$= - \frac{1}{2\pi} \left[(r_w - r) \int_0^{\pi} \frac{d\theta'}{R^3} + r \int_0^{\pi} \frac{1 - \cos\theta'}{R^3} \, d\theta' \right]$$

$$= -\frac{1}{2\pi}\left[(r_w - r)\frac{2E(\pi/2, k)}{[(x-x')^2+(r+r_w)^2]^{1/2}[(x-x')^2+(r-r_w)^2]}\right.$$

$$\left. + \frac{1}{r_w}\frac{F(\pi/2, k) - E(\pi/2, k)}{[(x-x')^2 + (r+r_w)^2]^{1/2}}\right] \tag{A-3}$$

(see 291.01 and 291.06 of ref. [107]).

Hence the expression (A-1) leads to

$$I_{S_w} = \int_{-\infty}^{\infty}\left[(r_w\frac{r_w - r}{\pi}\frac{E(\pi/2, k)}{[(x-x')^2 + (r_w-r)^2][(x-x')^2 + (r+r_w)^2]^{1/2}}\right.$$

$$\left. + \frac{1}{2\pi}\frac{F(\pi/2, k) - E(\pi/2, k)}{[(x-x')^2 + (r+r_w)^2]^{1/2}})\phi(x', r_w)\right.$$

$$\left. - \frac{p}{\beta}\frac{r_w}{\pi}\frac{F(\pi/2, k)}{[(x-x')^2 + (r+r_w)^2]^{1/2}}\phi_{x'}(x', r_w)\right] dx' \tag{A-4}$$

(ii) Differentiation of the Wall Term in Eq. (2.3.10) 2ith
Respect to x

The wall integral term in Eq. (2.3.10) is

$$I_{S_w} = \int_{-\infty}^{\infty}\left[(A + B)\phi(x', r_w) - C\frac{p}{\beta}\phi_{x'}(x', r_w)\right] dx' \tag{A-5}$$

Differentiating with respect to x we get,

$$\frac{\partial I_{S_w}}{\partial x} = \int_{-\infty}^{\infty}\left[\phi\frac{\partial}{\partial x}(A + B) - \frac{p}{\beta}\phi_{x'}\frac{\partial C}{\partial x}\right] dx'$$

$$= -\int_{-\infty}^{\infty}\left[\phi\frac{\partial}{\partial x'}(A + B) - \frac{p}{\beta}\phi_{x'}\frac{\partial C}{\partial x'}\right] dx'$$

$$\left(\text{since }\frac{\partial}{\partial x} = -\frac{\partial}{\partial x'}\right)$$

From 710.00 of ref. [107]

$$\frac{\partial C}{\partial x'} = \frac{\partial}{\partial x'} \left[\frac{r_w}{\pi} \frac{F(\pi/2, k)}{\{(x - x')^2 + (r + r_w)^2\}^{1/2}} \right]$$

$$= \frac{r_w}{\pi} \frac{(x - x') E(\pi/2, k)}{[(x - x')^2 + (r_w - r)^2] [(x - x')^2 + (r_w + r)^2]^{1/2}}$$

Hence

$$\frac{\partial I_{S_w}}{\partial x} = - \left[\phi(x', r_w)(A + B) \Big|_{-\infty}^{+\infty} - \int_{-\infty}^{\infty} \phi_{x'} (A + B) \, dx' \right.$$

$$\left. - \frac{r_w}{\pi} \frac{p}{\beta} \int_{-\infty}^{\infty} \frac{(x-x') E(\pi/2, k) \phi_{x'}(x', r_w)}{[(x-x')^2+(r_w-r)^2][(x-x')^2+(r+r_w)^2]^{1/2}} dx' \right]$$

The first term on the R.H.S. vanishes because of the
wall B.C. (2.2.16b), and the last two terms can be combined to
yield

$$\frac{\partial I_{S_w}}{\partial x} = \int_{-\infty}^{\infty} (A + B + D) \, u(x', r_w) \, dx' \qquad \text{(A-6)}$$

where,

$$D = - \frac{p}{\beta} \frac{r_w}{\pi} \frac{(x-x') E(\pi/2, k)}{[(x-x')^2 + (r_w-r)^2] [(x-x')^2 + (r+r_w)^2]^{1/2}}$$

APPENDIX B

INFLUENCE CO-EFFICIENTS FOR AXISYMMETRIC FLOW

(i) Derivation of the Field Influence Coefficient $a_F^{kl}(x,r)$

From Eq. (2.4.3)

$$a_F^{kl}(x,r) = \int_{x_k-\delta}^{x_k+\delta} dx' \int_{r_1-h}^{r_1+h} r' \, dr' \int_0^{2\pi} \frac{\partial \Psi}{\partial x'} \, d\theta'$$

Carrying out integration with respect to x' first

$$a_F^{kl}(x,r) = \frac{1}{4\pi} \int_{r_1-h}^{r_1+h} r' \, dr' \int_0^{2\pi} d\theta' \left[\frac{1}{(x_1^2+r'^2+r^2-2rr'\cos\theta')^{1/2}} \right.$$

$$\left. - \frac{1}{(x_2^2+r'^2+r^2-2rr'\cos\theta')^{1/2}} \right]$$

where

$$X_1 = x - x_k - \delta \qquad \text{and} \qquad X_2 = x - x_k + \delta$$

Denoting by Δ the difference in the value of the integrand for the arguments X_1 and X_2

$$a_F^{kl}(x,r) = \frac{1}{4\pi} \int_{r_1-h}^{r_1+h} r' \, dr' \int_0^{2\pi} \left[\frac{1}{(x^2+r'^2+r^2-2rr'\cos\theta')^{1/2}} \right] d\theta'$$

$$= \frac{1}{4\pi} \int_{r_1-h}^{r_1+h} r' \, dr' \, \Delta \left[2g \, F(k) \right] \qquad \begin{array}{l} \text{See 291.00 of} \\ \text{ref. [107].} \end{array}$$

where F is the complete elliptic integral of the first kind and

$$g = \frac{2}{[x^2 + (r+r')^2]^{1/2}} \qquad k^2 = \frac{4rr'}{x^2 + (r+r')^2}$$

The elliptic integral is now treated as a constant and taken out of the integral sign with its value at the centre of the element. Therefore,

$$a_F^{kl}(x,r) = \frac{1}{4\pi} \Delta\left[2F(k) \int_{r_1-h}^{r_1+h} \frac{2r' \, dr'}{\{x^2 + (r+r')^2\}^{1/2}}\right]$$

$$= \frac{1}{\pi}\left[F(k_1) \, G(X_1) - F(k_2) \, G(X_2)\right] \qquad \text{(B-1)}$$

where

$$G(X) = (X^2 + R_1^2)^{1/2} - (X^2 + R_2^2)^{1/2} - r \ln \frac{R_2 + (X^2 + R_2^2)^{1/2}}{R_1 + (X^2 + R_1^2)^{1/2}}$$

$$\text{(B-2)}$$

$$X_1 = x - x_k - \delta \qquad R_1 = r + r_1 + h \qquad k^2 = \frac{4rr_1}{x^2 + (r+r_1)^2}$$

$$\} \, 1 \neq 1$$

$$X_2 = x - x_k + \delta \qquad R_2 = r + r_1 - h \qquad k_\nu = k(X_\nu) \quad \nu = 1,2$$

$$R_1 = r + h \qquad R_2 = r \qquad k^2 = \frac{r_2 r}{x^2 + (r + 0.25 \, r_2)^2} \qquad \text{for } 1 = 1.$$

(ii) Derivation of the Wall Influence Co-efficient $a_w^k(x,r)$

From Eq. (2.4.4)

$$a_w^k(x,r) = \int_{x_k-\delta}^{x_k+\delta} (A + B + D) \, dx'$$

The wall co-efficients consist of a symmetric point arising from the terms A and B and an anti-symmetric part arising from D. Treating the elliptic integrals as constant over each of the small elementary intervals of integration, the resulting simple integrals may be easily evaluated. Denoting

the symmetric part by a_S and anti-symmetric part by a_A the
integrations are carried out to obtain

for $r \neq r_w$ or $x \neq x_k$

$$a_S^k(x,r) = -\frac{r_w}{\pi} (\tan^{-1} \frac{X_2}{r - r_w} - \tan^{-1} \frac{X_1}{r - r_w}) \frac{E(k)}{\sqrt{d}}$$

$$+ \frac{1}{2\pi} (F(k) - E(k)) \ln \frac{X_2 + [X_2^2 + (r + r_w)^2]^{1/2}}{X_1 + [X_1^2 + (r + r_w)^2]^{1/2}}$$

(B-3)

$$a_A^k(x,r) = \frac{r_w}{2\pi} \frac{p}{\beta} \frac{E(k)}{\sqrt{d}} \ln \frac{X_2^2 + (r - r_w)^2}{X_1^2 + (r - r_w)^2}$$

(B-4)

$$d = (x - x_k)^2 + (r + r_w)^2 \qquad b = 4rr_w \qquad k = (b/d)^{1/2}$$

For $x = x_k$ and $r = r_w$ the integral involving $F(k)$ in the
expression for a_S becomes singular. It is therefore separately
evaluated as follows.

$$I_B = \frac{1}{2\pi} \int_{x_k-\delta}^{x_k+\delta} \frac{F(k)}{[(x - x')^2 + \gamma^2]^{1/2}} \, dx'$$

where $\gamma = 4r_w^2$

Putting $z = x_k - x'$ we get

$$I_B = \frac{1}{2\pi} \int_{-\delta}^{+\delta} \frac{F(k)}{(z^2 + \gamma^2)^{1/2}} \, dz$$

Since the integrand is even

$$I_B = \frac{1}{\pi} \int_0^\delta \frac{F(k)}{(z^2 + \gamma^2)^{1/2}} \, dz$$

Since $z/\gamma \ll 1$ we can write,

$$I_B = \frac{1}{\pi\gamma} \int_0^\delta F(k)\ dz$$

$$= \frac{1}{\pi\gamma} \int_0^\delta [(a_o + a_1 k_1^2 + a_2 k_1^4) - (b_o + b_1 k_1^2 + b_2 k_1^4)\ \ln k_1^2\ dz]$$

(see ref. [108])

where $\quad k_1^2 = z^2/(z^2 + \gamma^2) \simeq z^2/\gamma^2$

Neglecting all terms higher than first order and integrating term by term we obtain,

$$I_B = \frac{1}{\pi}\left[a_o \delta^* + \frac{a_1}{3}\delta^{*3} + \frac{a_2}{5}\delta^{*5} + 2(\ln\delta^* - 1)\{b_o\ \delta^* + \frac{b_1}{3}\delta^{*3}\right.$$

$$\left. + \frac{b_2}{5}\delta^{*5}\}\ \right] \qquad\qquad (B-5)$$

where $\quad \delta^* = \delta/\gamma$.

Hence for $\underline{x = x_k}$ and $\underline{r = r_w}$ we may write

$$a_S^k(x,r) = I_B - \frac{1}{2\pi}\ E(k)\ \ln\frac{X_2 + [X_2^2 + (r + r_w)^2]^{1/2}}{X_1 + [X_1^2 + (r + r_w)^2]^{1/2}} \qquad (B-6)$$

The co-efficients a_o, a_1 etc. and b_o, b_1 etc. are [108]

$a_o = 1.3862944 \qquad a_1 = 0.1119723 \qquad a_2 = 0.0725296$

$b_o = 0.5 \qquad\qquad b_1 = 0.1213478 \qquad b_2 = 0.028879.$

The wall influence co-efficient is obtained from

$$a_w(x,r) = a_S(x,r) + a_A(x,r) \qquad\qquad (B-7)$$

For a computational grid with uniform mesh width in the axial direction, the following relations of symmetry hold.

$$a_{F_{ij}}^{kl} = -a_{F_{kj}}^{il} \qquad\qquad a_{F_{ij}}^{kl} = a_{F_{i-1,j}}^{k-1,l}$$

Similarly

$$a_{S_{ij}}^{k} = a_{S_{kj}}^{i} \qquad \text{and} \qquad a_{A_{ij}}^{k} = - a_{A_{kj}}^{i}$$

$$a_{S_{ij}}^{k} = a_{S_{i-1,j}}^{k-1} \qquad \text{and} \qquad a_{A_{ij}}^{k} = a_{A_{i-1,j}}^{k-1}$$

Hence the storage area required for a_F is (k x l x j) location and for a_W it is (2 x k x j) locations.

APPENDIX C

EQUATIONS FOR SLENDER BODY GEOMETRY

The equation for the profile of a general parabolic body of revolution is given by the following two equations, ref. [96].

For body with maximum thickness forward of its mid-point

$$\overline{R}(x) = C \, R_{max} \left[1 - x - (1 - x)^n \right] \tag{C-1}$$

For body with maximum thickness aft of its mid-point

$$\overline{R}(x) = C \, R_{max} (x - x^n) \tag{C-2}$$

The following table gives the data for various C and n combinations with locations of maximum thickness.

x at R_{max}	C	n	Formula used	Sting
0.3	1.71	6.03	C-1	0.71
0.4	2.36	3.39	C-1	0.78
0.5	4.00	2.00	C-1 or C-2	0.85
0.6	2.36	3.39	C-2	0.89
0.7	1.71	6.03	C-2	0.93

The cross-sectional area function $s(x)$ and its derivatives are

$$s(x) = \pi R^2(x)$$

$$s'(x) = 2 \pi R R'(x)$$

$$s''(x) = 2 \pi (R R''(x) + R'^2(x))$$

CALCULATION OF ARTIFICIAL VISCOSITY FOR PLANE FLOW

Simplification of artificial viscosity term T, Eq. (3.2.18)

$$T = -\frac{\partial}{\partial s}\left[\Delta s\ \mu\ \frac{\rho}{c^2}\ Q\ \frac{\partial}{\partial s}\left(\frac{Q^2}{2}\right)\right] \qquad (3.2.18)$$

$$= -\frac{\partial}{\partial s}\left[\Delta s\ \mu\ \frac{\rho}{c^2}\ \left\{U\ \frac{\partial}{\partial x}\left(\frac{U^2 + V^2}{2}\right) + V\ \frac{\partial}{\partial y}\left(\frac{U^2 + V^2}{2}\right)\right\}\right]$$

$$= -\frac{\partial}{\partial s}\left[\Delta s\ \mu\ \frac{\rho}{c^2}\ \left\{(U^2 U_x + UV\ V_x) + (V^2 V_y + UV\ U_y)\right\}\right]$$

$$= -\frac{1}{Q}\left(U\ \frac{\partial}{\partial x} + V\ \frac{\partial}{\partial y}\right)\left[\frac{Q}{U}\ \Delta x\ \mu\ \frac{\rho}{c^2}\ (U^2 U_x + UV\ V_x)\right.$$

$$\left. + \frac{Q}{V}\ \Delta y\ \mu\ \frac{\rho}{c^2}\ (V^2 V_y + UV\ U_y)\right] \qquad (D\text{-}1)$$

Inspection of Eq. (D-1) shows that its highest derivative terms are the same as those in Eq. (3.2.15). Neglecting the cross derivative terms in (D-1) we obtain

$$T\ \simeq\ \frac{\partial G}{\partial x} + \frac{\partial H}{\partial y} \qquad (D\text{-}2)$$

where $\quad G = -\ \Delta x\ \mu\ \frac{\rho}{c^2}\ (U^2 U_x + UV\ V_x)$

and $\quad H = -\ \Delta y\ \mu\ \frac{\rho}{c^2}\ (V^2 V_y + VU\ U_y)$

CALCULATION OF FIELD INFLUENCE CO-EFFICIENTS FOR PLANE FLOW

Influence Co-efficients for a Plane Field Source Element
of Unit Strength

Figure (E-1) shows an arbitrary, plane, quadrilateral
source element S of unit strength. The co-ordinates are
referred to axes fixed at the grid point of the element whose
corners are numbered 1, 2, 3 and 4, going clockwise. The
co-ordinates of this grid point in the global co-ordinate
system fixed at the leading edge of the airfoil are (ξ_{kl}, η_{kl}).
The u and v components of velocity, in the x and y direction
respectively, induced by S at an arbitrary field point (x, y)
are given by

$$u_S(x,y) \;=\; e^{kl}(x,y) \;=\; \frac{1}{2\pi} \iint_S \frac{x - x'}{(x-x')^2 + (y-y')^2} \, dx' \, dy'$$

$$(E-1)$$

$$v_S(x,y) \;=\; f^{kl}(x,y) \;=\; \frac{1}{2\pi} \iint_S \frac{y - y'}{(x-x')^2 + (y-y')^2} \, dx' \, dy'$$

$$(E-2)$$

The integrals may be conveniently evaluated by writing
each as a sum of integrals over 4 infinite strips. First we
consider the evaluation of the integral in Eq. (E-1).

Consider the side 1-2 of the quadrilateral element
1-2-3-4-1 shown in Figure (E·1). This side is now assumed to
be replaced by a source sheet of infinite extent with sides
parallel to the x-axis as shown in Figure (E-2). Assuming
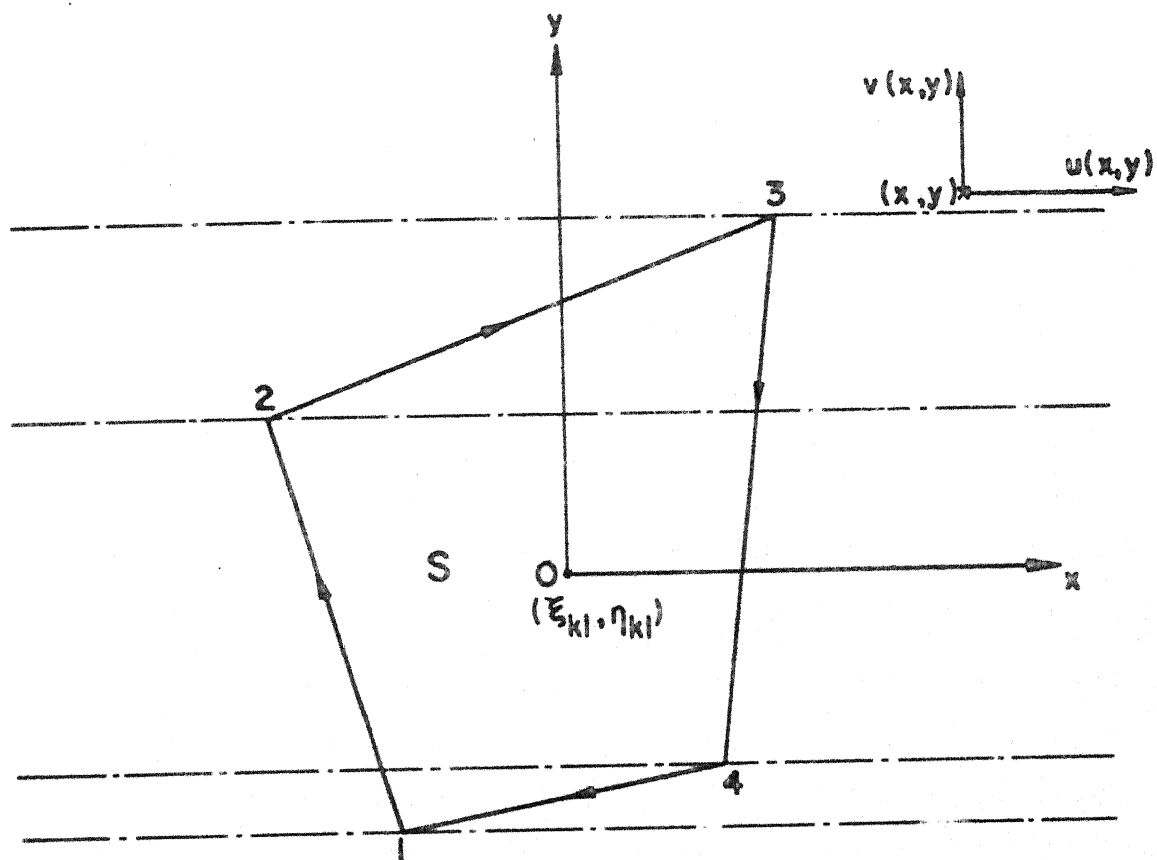
Fig. E-1
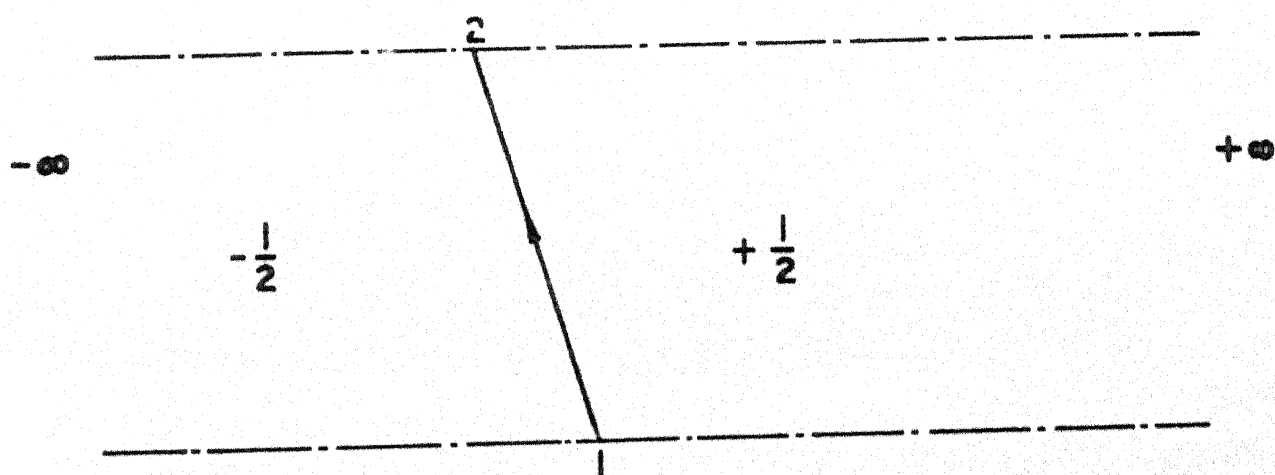
Fig. E-2. Influence coefficient calculation for a plane field source element

that the quadrilateral is traversed in the clockwise direction as indicated in Figure (E-1), we assign a uniform source strength -1/2 for the semi-infinite sheet lying to the left of side 1-2 and +1/2 for the semi-infinite sheet lying to its right.

Traversing the quadrilateral clockwise, the other sides are also replaced by similar infinite source strips with source strengths +1/2 or -1/2 according as the semi-infinite part of the strip lies to the right or left of the side being traversed. Thus one infinite strip is associated with each side of the quadrilateral and the source strength undergoes a jump of unit strength across the side. These source strips may be denoted by $S_{\mu\nu}$ where

$$\nu = \mu + 1 \qquad \text{for} \quad \mu = 1, 2, 3$$
$$= 1 \qquad \text{for} \quad \mu = 4$$

If $\sigma_{\mu\nu}$ denotes the source strength function having values +1/2 and -1/2 on either side of the line $\mu - \nu$, we may write the integral in Eq. (E-1) as

$$e^{kl}(x,y) = \frac{1}{2\pi} \sum_{\mu=1}^{4} \iint_{S_{\mu\nu}} \frac{x - x'}{(x-x')^2 + (y-y')^2} \sigma_{\mu\nu} \, dx' \, dy'$$

$$= \sum_{\mu=1}^{4} e_{\mu\nu}^{kl}(x,y) \qquad\qquad (E-3)$$

Each integral in Eq. (E-3) may be explicitly evaluated. For example, consider the integral over the strip $S_{\mu\nu}$. The equation for the line joining $\mu - \nu$ is given by

$$\frac{y' - y'_\nu}{y'_\mu - y'_\nu} = \frac{x' - x'_\nu}{x'_\mu - x'_\nu} \qquad\qquad (E-4)$$

so that

$$x' = p_{\mu\nu} + q_{\mu\nu} y' \tag{E-5}$$

where

$$q_{\mu\nu} = \frac{x'_\nu - x'_\mu}{y'_\nu - y'_\mu} \tag{E-6}$$

$$p_{\mu\nu} = - q_{\mu\nu} y'_\nu + x'_\nu \tag{E-7}$$

The integral for $e_{\mu\nu}^{kl}$ can be written as

$$e_{\mu\nu}^{kl}(x,y) = -\frac{1}{4\pi} \int_{y'_\mu}^{y'_\nu} dy' \{ -\frac{1}{2} \int_{-\infty}^{x'(y')} + \frac{1}{2} \int_{x'(y')}^{\infty} \} \frac{\partial}{\partial x'} \ln$$

$$[(x-x')^2 + (y-y')^2] \, dx' \, dy' \tag{E-8}$$

where $x'(y')$ is given by Eq. (E-5). Eq. (E-8) holds for both $y'_\mu > y'_\nu$ and $y'_\mu < y'_\nu$.

The integral in Eq. (E-8) can be easily evaluated in terms of elementary functions and the following expression may be obtained

$$e_{\mu\nu}^{kl}(x,y) = \frac{1}{4\pi} [(y'_\nu - y'_\mu)\ln h_3^2 + F(y'_\nu, y'_\mu, h_2, h_4)] \tag{E-9}$$

where

$$F(\gamma, \delta, \alpha, \beta) = (\gamma + \alpha) \ln [(\gamma + \alpha)^2 + \beta^2] -$$

$$(\delta + \alpha) \ln [(\delta + \alpha)^2 + \beta^2] - 2(\gamma - \delta) +$$

$$2\beta (\tan^{-1} \frac{\gamma + \alpha}{\beta} - \tan^{-1} \frac{\delta + \alpha}{\beta}) \tag{E-10}$$

Dropping the subscripts $\mu\nu$, the quantities h_1, h_2, h_3 and h_4 are obtained from

$$h_1^2 = [(x - p)^2 + y^2]/h_3^2 \tag{E-11}$$

$$h_2 = (x - p)q + y /h_3^2 \tag{E-12}$$

$$h_3^2 = 1 + q^2 \tag{E-13}$$

$$h_4^2 = h_1^2 - h_2^2 \tag{E-14}$$

The calculation of the influence co-efficients for the v-component proceeds in the same fashion except that in this case, the strips have sides parallel to the y-axis. Replacing y' by x', x by y and vice versa we get,

$$f_{\mu\nu}^{kl}(x,y) = -\frac{1}{4\pi}[(x_\nu' - x_\mu')\ln g_3^2 + F(x_\nu', x_\mu', g_2, g_4)] \tag{E-15}$$

and

$$f^{kl}(x,y) = \sum_{\mu=1}^{4} f_{\mu\nu}^{kl}(x,y) \tag{E-16}$$

The corresponding expressions for g_1, g_2 etc. are as follows.

$$\frac{y' - y_\nu'}{y_\nu' - y_\mu'} = \frac{x' - x_\nu'}{x_\nu' - x_\mu'} \tag{E-17}$$

$$y' = r_{\mu\nu} + s_{\mu\nu} x' \tag{E-18}$$

$$s_{\mu\nu} = \frac{y_\nu' - y_\mu'}{x_\nu' - x_\mu'} \tag{E-19}$$

$$r_{\mu\nu} = -s_{\mu\nu} x_\nu' + y_\nu' \tag{E-20}$$

$$g_1^2 = [x^2 + (y - r)^2]/g_3^2 \tag{E-21}$$

$$g_2 = [x + (y - r)s]/g_3^2 \tag{E-22}$$

$$g_3^2 = 1 + s^2 \tag{E-23}$$

$$g_4^2 = g_1^2 - g_2^2 \qquad\qquad \text{(E-24)}$$

The function F is the same as defined by Eq. (E-10).

Co-efficients calculated using Eqs. (E-1) to (E-24), though exact, involve large number of function evaluations. When the plane source element is far from the field point it can be approximated by a point source and the influence co-efficients can be evaluated from the formulae

$$e^{kl}(x,y) = \frac{1}{2\pi} \frac{x - x'}{(x - x')^2 + (y - y')^2} \Delta_{kl} \qquad\qquad \text{(E-25)}$$

$$f^{kl}(x,y) = \frac{1}{2\pi} \frac{y - y'}{(x - x')^2 + (y - y')^2} \Delta_{kl} \qquad\qquad \text{(E-26)}$$

where Δ_{kl} represents the area of the quadrilateral element k,l. Formulae (E-25) and (E-26) were used when the condition

$$[(x - x')^2 + (y - y')^2]^{1/2} > 4 D_{kl}$$

was satisfied, with D_{kl} denoting the length of the longer of the two diagonals of the source element. In actual calculations it was found that about 40 percent of the total number of influence co-efficients could be calculated from Eqs. (E-25) and (E-26).

The co-efficients e and f calculated as above remain the same for the airfoil fixed co-ordinate system since the induced velocities u and v are invariant with respect to simple translation of the origin.

To economise on storage for special cases, the following symmetries in the influence co-efficients may be noted.

(a)   Flow Past a Symmetric Airfoil at Zero Angle of Attack

In this case for every plane source element S on the upper plane, there corresponds a conjugate element S^* on the lower plane. The geometry of S^* is the mirror reflection of S. Because of symmetry, the calculation for such a flow needs to be carried out only on one half of the plane. Hence the velocity components induced by S^* at a field point on the upper half plane must also be taken into account. Since the induced velocities for two source elements conjugate to each other are symmetric in u and anti-symmetric in v, we have,

$$u_{S^*}(x, y) = u_S(x, -y) \qquad \text{(E-27)}$$

$$v_{S^*}(x, y) = -v_S(x, -y) \qquad \text{(E-28)}$$

so that the total induced velocities at (x, y) are

$$u(x, y) = u_S(x, y) + u_{S^*}(x, -y) \qquad \text{(E-29)}$$

$$v(x, y) = v_S(x, y) - v_{S^*}(x, -y) \qquad \text{(E-30)}$$

(b)   Lifting Flow Past Symmetric Airfoils

When the airfoil is symmetric, the computational grid on the physical plane is symmetric about the x-axis. It is then obvious that given a source element S on the upper half plane, there corresponds to it a conjugate element S^* on the lower half plane. The velocity components induced at a point (x, y) by S^* of unit strength will therefore satisfy Eqs. (E-27) and (E-28). This implies then that we need to

store only the influence co-efficients of the source elements
on the upper half plane. These must however be evaluated for
all the grid points both on the upper and lower half planes.

For general flow about an arbitrary cambered airfoil,
neither of these symmetries hold. Hence for this case, the
entire array of influence co-efficients have to be computed
and stored.

APPENDIX F

INFLUENCE CO-EFFICIENTS FOR THE INTERNAL SOURCE AND VORTICITY DISTRIBUTIONS

(i) Uniform Source Distribution

Referring to Figure (F-1), we consider a source element k of width δ and choose co-ordinate axes with origin at the mid-point of the element and x-axis coinciding with the element. Bars below refer to the element fixed co-ordinate system.

The velocity components induced at a field point $(\bar{x}, \bar{y})$ by a source element of unit strength are

$$\bar{u}(\bar{x}, \bar{y}) = \frac{1}{2\pi} \int_{-\delta}^{+\delta} \frac{\bar{x} - \bar{\xi}}{(\bar{x} - \bar{\xi})^2 + \bar{y}^2} \, d\bar{\xi} \qquad (F-1)$$

$$\bar{v}(\bar{x}, \bar{y}) = \frac{1}{2\pi} \int_{-\delta}^{+\delta} \frac{\bar{y}}{(\bar{x} - \bar{\xi})^2 + \bar{y}^2} \, d\bar{\xi} \qquad (F-2)$$

Carrying out the integrations we obtain,

$$\bar{u}(\bar{x}, \bar{y}) = \frac{1}{4\pi}\ln \frac{(\bar{x} + \delta)^2 + \bar{y}^2}{(\bar{x} - \delta)^2 + \bar{y}^2} = F(\bar{x}, \bar{y}, \delta) \qquad (F-3)$$

$$\bar{v}(\bar{x}, \bar{y}) = \frac{1}{2\pi} \left(\tan^{-1} \frac{\bar{x} + \delta}{\bar{y}} - \tan^{-1} \frac{\bar{x} - \delta}{\bar{y}} \right) = G(\bar{x}, \bar{y}, \delta) \qquad (F-4)$$

If the element inclination to the x-axis of the air-foil fixed co-ordinate system is θ and the co-ordinates of its mid-point are (x_o, y_o), we obtain the influence co-efficients in the global co-ordinate system as follows.

$$\bar{x} = (x - x_o) \cos\theta + (y - y_o) \sin\theta \qquad (F-5)$$
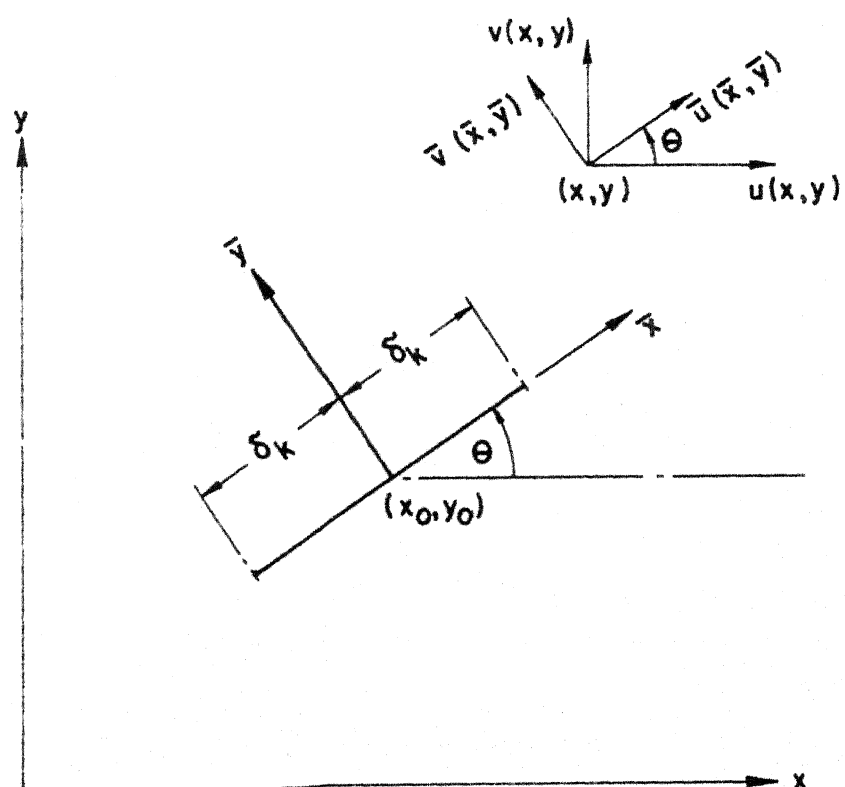
Fig. F-1. Influence co-efficient calculation for the line elements

$$\bar{y} = - (x - x_o) \sin\theta + (y - y_o) \cos\theta \qquad \text{(F-6)}$$

$$a^k(x, y) = F(\bar{x}, \bar{y}, \delta)\cos\theta - G(\bar{x}, \bar{y}, \delta)\sin\theta \qquad \text{(F-7)}$$

$$b^k(x, y) = F(\bar{x}, \bar{y}, \delta)\sin\theta + G(\bar{x}, \bar{y}, \delta)\cos\theta \qquad \text{(F-8)}$$

where x and y are co-ordinates of the field point in the air-foil fixed (global) co-ordinate system.

(ii) Linear Vorticity Distribution

A linear variation of the vorticity strength is assumed over the element k in terms of the values at its end points.

Again assuming an element fixed co-ordinate system,

$$\gamma(x) = \frac{\gamma_{k+1} + \gamma_k}{2} - \frac{\gamma_{k+1} - \gamma_k}{2\delta} x \qquad \text{(F-9)}$$

The velocity components induced at a field point $(\bar{x}, \bar{y})$ are given by

$$\bar{u}(\bar{x}, \bar{y}) = \frac{1}{2\pi} \int_{-\delta}^{+\delta} \frac{\bar{y}}{(\bar{x} - \bar{\xi})^2 + \bar{y}^2} \gamma(\bar{\xi}) \, d\bar{\xi} \qquad \text{(F-10)}$$

$$\bar{v}(\bar{x}, \bar{y}) = - \frac{1}{2\pi} \int_{-\delta}^{+\delta} \frac{\bar{x} - \bar{\xi}}{(\bar{x} - \bar{\xi})^2 + \bar{y}^2} \gamma(\bar{\xi}) \, d\bar{\xi} \qquad \text{(F-11)}$$

Substituting Eq. (F-9) in (F-10) and (F-11) carrying out the integrations and transforming to the global co-ordinate system, the influence co-efficients c and d may be obtained as (n below refers to the number of mean line elements)

$$c^k(x, y) = A^{k-1}(x, y) + B^k(x, y) \qquad \text{(F-12)}$$

$$d^k(x, y) = c^{k-1}(x, y) + D^k(x, y) \tag{F-13}$$

$$c^1(x, y) = B^1(x, y) \tag{F-14}$$

$$c^{n+1}(x, y) = A^n(x, y) \tag{F-15}$$

$$d^1(x, y) = D^1(x, y) \tag{F-16}$$

$$d^{n+1}(x, y) = c^n(x, y) \tag{F-17}$$

$$A^k(x, y) = \frac{1}{2}\left[\left(G + \frac{yF}{\delta} - \frac{xG}{\delta}\right)\cos\theta + \left(F - \frac{xF}{\delta} - \frac{yG}{\delta} + \frac{1}{2\pi}\right)\sin\theta\right] \tag{F-18}$$

$$B^k(x, y) = \frac{1}{2}\left[\left(G - \frac{yF}{\delta} + \frac{xG}{\delta}\right)\cos\theta + \left(F + \frac{xF}{\delta} + \frac{yG}{\delta} - \frac{1}{2\pi}\right)\sin\theta\right] \tag{F-19}$$

$$C^k(x, y) = \frac{1}{2}\left[\left(G + \frac{yF}{\delta} - \frac{xG}{\delta}\right)\sin\theta - \left(F - \frac{xF}{\delta} - \frac{yG}{\delta} + \frac{1}{2\pi}\right)\cos\theta\right] \tag{F-20}$$

$$D^k(x, y) = \frac{1}{2}\left[\left(G - \frac{yF}{\delta} + \frac{xG}{\delta}\right)\sin\theta - \left(F + \frac{xF}{\delta} + \frac{yG}{\delta} - \frac{1}{2\pi}\right)\cos\theta\right] \tag{F-21}$$

where (x, y) refer to the co-ordinates of the field point while F and G are functions defined by Eqs. (F-3) and (F-4) with arguments given by Eqs. (F-5) and (F-6).

(iii) Point Sources

The velocities induced by the point sources of unit strength may be included in the co-efficients a^k and b^k. We have

$$a^k(x, y) = \frac{1}{2\pi}\frac{x - x_k}{(x - x_k)^2 + (y - y_k)^2} \qquad k = n+1,\ n+2 \tag{F-22}$$

$$b^k(x, y) \;=\; \frac{1}{2\pi}\frac{y - y_k}{(x - x_k)^2 + (y - y_k)^2} \qquad k = n+1, \; n+2 \qquad \text{(F-23)}$$

(iv) Point Vortex

The induced velocities due to a point vortex located at (x_k, y_k) may be included in the co-efficients c^k and d^k.

$$c^k(x, y) \;=\; \frac{1}{2\pi}\frac{y - y_k}{(x - x_k)^2 + (y - y_k)^2} \qquad \text{(F-24)}$$

$$k = n+2$$

$$d^k(x, y) \;=\; -\frac{1}{2\pi}\frac{x - x_k}{(x - x_k)^2 + (y - y_k)^2} \qquad \text{(F-25)}$$

APPENDIX G

LISTING OF THE PROGRAM FOR PLANE, TWO-DIMENSIONAL
FULL-POTENTIAL PROBLEM

```
C       THIS IS THE LATEST VERSION OF THE PROGRAM.
C       BEFORE TAKING THE LISTING TRY TO CARRY OUT ONE SAMPLE EXECUTION.
C
C       This program should be executed together with
C       AERFOL.FOR,GRID.FOR,AUFAVE.FOR and ARTVS1.FOR
C        and also linked to NAG library. Before executing
C       this program the field aerodynamic coefficients FSU and FSV
C       must be computed by executing GRID.FOR with HESS.FOR,AERFOL.FOR
C       These coefficients are output on the disk in the files
C       FOR22.DAT and FOR23.DAT. The main program is now ready
C       for execution.
C
C       This program computes lifting compressible flow over airfoils.
C       Convergence of the iteration scheme may be expected for maximum
C       local MACH numbers up to 1.4. The input data format as well as
C       their meaning is explained in the data file B1.DAT. User
C       interaction becomes necessary when the iteration limit set by
C       ITMAX is exceeded. Appropriate messages are given at such
C       times and user may respond according to his choice.
C       At convergence similar messages are displayed on the terminal
C       and user responds with appropriate inputs.
C
C       this is the MAIN program that includes subroutines UVFSIG,UVFGF2
C       FRCCAL,DERCOF,DERIV and FORCF. The rest of the subroutines
C       required come from the routines mentioned above. The output is
C       written on the disk in the file FOR21.DAT. The subroutine
C       F04AAF is a NAG routine used for matrix inversion.
C
C       The subroutine AERFOL.FOR has its own data file for input.This
C       data file AFL.DAT contains  data pertaining to the airfoil.
C
C       PROGRAM DEVELOPED AND VALIDATED BY: K.S.RAVICHANDRAN
C
C       To the best of the authors knowledge the program is free of
C       bugs for the airfoils for which calculations were carried out
C       reported in this thesis.However the program has not been
C       thouroughly checked for all the options.In particular the
C       subroutine AERFOL has not been test run for all possible ways
C       in which the aerfoil coordinates may be prescribed.Therefore
C       for some of the cases not tried so far bugs will very likely
C       be found in this subroutine.
C
C       DATE:JUNE 10, 1983
C       *****************************************************************
        DIMENSION XSU(55),YSU(55)
        DIMENSION WKAREA(100)
        DIMENSION XCH(55),YCH(55),CPCH(55)
        DIMENSION ZETA(55)
        COMPLEX ZGRID,ZNET
        DIMENSION CPU(55),CPL(55),CPLINU(55),CPLINL(55)
        DIMENSION CPLIN(55),CP(55)
        DIMENSION CC(50,50),CCINV(50,50)
C       *****************************************************************
        COMMON /E1/ NLS,NMAX,MMAX,NW,NC,ICAM,IALFA
        COMMON /E2/ FSM,ALFA,GAM,EXPGAM,FGAM,PRF,QSONIC,SK,BETA2,LAMAX
        COMMON /E3/ RADLE,IAFCP1,IAFCP2,TAU,BT(10),CT(10)
        COMMON /E4/ CT1,CT2,CT3,CT4,CT5
        COMMON /E5/ CSALFA,SNALFA
        COMMON /C1/ XS(55),YS(55),YPS(55),DS(55),SINT(55),COST(55)
        COMMON /C2/ XC(55),YC(55),YPC(55),DSP(55)
        COMMON /C3/ AS(23,51,5),BS(23,51,5),CV(23,51,5),DV(23,51,5)
        COMMON /C4/ UFP(51,5),VFP(51,5)
        COMMON /C5/ FSV(51,26,5),FSU(51,5,26,5)
        COMMON /C6/ ZGRID(52,6),ZNET(52,6)
        COMMON /C7/ FFC(55),FPCP(55)
        COMMON /C8/ RHO(51,5),RHOX(51,5),RHOY(51,5)
        COMMON /C10/ UF(51,5),VF(51,5)
        COMMON /C11/ UFX(51,5),UFY(51,5),VFX(51,5),VFY(51,5)
        COMMON /C12/ Q(51,5),QS(51,5),RHO(51,5)
        COMMON /C13/ UFNL(51,5),VFNL(51,5)
        COMMON /C14/ GFN(51,5),R(51,5),GF(51,5)
        COMMON /C15/ SIG(55),GAMA(55)
        COMMON /C19/ A2(51,5),RMACH(51,5)
```

```
      COMMON /ZNT/ Z(51,5),DZX(51,5),DZY(51,5)
      COMMON /CRT/ DELXI(51,5)
      LOGICAL FLAG,AVRG
      REAL LAMAX
C
C     Read input data
C
C     -----------------------------------------------------------------
      OPEN(UNIT=1,DEVICE='DSK',FILE='B1.DAT')
      READ(1,*)AVRG
      READ(1,*)FSM,ALFA,GAM,LAMAX
      READ(1,*)NBLS,NILS,NALS,NW,NJ,JC,SFJ,NJLAST,BETMIN,DELX,
     1 XSHL,XSHR,DELTA,EPSTE
      READ(1,*)EPS,ITMAX,ALP1,ALP2
C     -----------------------------------------------------------------
      WRITE(20,701)
701   FORMAT(/10X,70(1H*)/)
      WRITE(20,601)
601   FORMAT(15X,'TWO-DIMENSIONAL TRANSONIC FLOW PAST PLANE AIRFOILS'/
     115X,'USING FULL-POTENTIAL EQUATION AND EXACT TANGENCY BOUNDARY'/
     221X,'CONDITIONS BY THE INTEGRAL EQUATION METHOD.'////
     320X,'The following is the input data specification')
      WRITE(20,701)
      WRITE(20,602)AVRG
602   FORMAT(15X,'AVRG=',L3)
      WRITE(20,603)FSM,ALFA,GAM,LAMAX
603   FORMAT(/15X'FSM=',F6.3,3X,'ALFA=',F6.3,3X,'GAM=',F6.3,3X,'LAMAX=
     1',F6.3/)
      WRITE(20,604)NBLS,NILS,NALS,NW
604   FORMAT(15X,'NBLS=',I3,5X,'NILS=',I3,5X,'NALS=',I3,5X,'NW=',I3/)
      WRITE(20,605)NJ,JC,SFJ,NJLAST
605   FORMAT(15X,'NJ=',I3,5X,'JC=',I3,5X,'SFJ=',F5.2,5X,'NJLAST=',I3/)
      WRITE(20,606)BETMIN,DELX,DELTA
606   FORMAT(15X,'BETMIN=',F6.3,5X,'DELX=',F7.4,5X,'DELTA=',F10.7/)
      WRITE(20,607)XSHL,XSHR,EPSTE
607   FORMAT(15X,'XSHL=',F6.3,5X,'XSHR=',F6.3,5X,'EPSTE=',F6.3/)
      WRITE(20,608)EPS,ITMAX,ALP1,ALP2
608   FORMAT(15X,'EPS=',F6.3,5X,'ITMAX=',I3,5X,'ALP1=',F6.3,5X,'ALP2='
     1,F6.3/)
      WRITE(20,701)
C
C     calculate the basic constants and parameters of the flow.
C
C     -----------------------------------------------------------------
      CALL RTIME(ISEG1)
      PIR=3.1415926535897932;   EXPGAM=1./(GAM-1.);   PIF=0.5/PIR
      EXPGAM=1./(GAM-1.);   FSM2=FSM**2;   FGAM=0.5*(GAM-1.)*FSM2
      PRF=2./GAM/FSM2;   AQ=0.5*(1.+GAM);   BQ=1./FSM2+0.5*(GAM-1.)
      QSONIC=SQRT(BQ/AQ);   ALFAD=ALFA;   IALFA=ALFA;   GAM1=1.+GAM
      CPSTAR=PRF*((1.-FGAM*(QSONIC**2-1.))**(GAM*EXPGAM)-1.0)
      ALFA=ALFA*PIR/180.;   CSALFA=COS(ALFA);   SNALFA=SIN(ALFA)
C     -----------------------------------------------------------------
C
C     set up the grid
C
      CALL GRID(NBLS,NILS,NALS,NW,NJ,JC,SFJ,NJLAST,BETMIN,ORIGIN
     1,DELX,EPSTE,XSHL,XSHR,ICAM,NMAX,MMAX,NLS,NC,DELTA,CHORD)
C
C     compute the coefficients for the derivative calculation
C
      CALL DERCOF
C
C     compute the influence coefficients for the internal singularitis
C
      CALL AUFAVF(NLS,NC,NXI,NJ,NMAX,MMAX,NJLAST,ICAM)
      IMID=(MMAX+1)/2
      XMAX=REAL(ZNET(1,1));   XMIN=REAL(ZNET(IMID,MMAX+1))
      YMAX=AIMAG(ZNET(1,MMAX+1))
      NBLC=NBLS+2;   NILC=NILS;   NALC=NALS
C     -----------------------------------------------------------------
      WRITE(20,501)
      IF(AVRG)WRITE(20,565);   IF(.NOT.AVRG)WRITE(20,566)
      WRITE(20,557)FOIL,VISFUN
      WRITE(20,503)FSM,TAU,ALFAD,LAMAX
      WRITE(20,504)NLS,NC
```

```fortran
      WRITE(2),507)HRLC,HALC,HILC,XSHL,XSHR
      WRITE(2),505)IMAX,MMAX,NW,YMAX,XMIN,VMAX
      WRITE(2),509)JSHLC,CPSTAR
      WRITE(2),701)
C-------------------------------------------------------------
      CALL RTIME(ISEG2)
      KMAX=IMAX;   IMAX=MMAX;   IF(ICAM.EQ.J)KMAX=(NMAX+1)/2
      OPEN(UNIT=22,DEVICE='DSK')
      OPEN(UNIT=23,DEVICE='DSK')
      DO  10   J=1,MMAX
      DO  11   I=1,IMAX
      READ(22,506)((FSU(I,J,K,L),K=1,KMAX),L=1,MMAX)
      IF(J.GT.1)GO TO 11
      READ(23,505)((FSV(I,K,L),K=1,KMAX),L=1,MMAX)
11    CONTINUE
      WRITE(5,888)J
10    CONTINUE
506   FORMAT(10F13.8)
888   FORMAT(2X,'input data read for J=',I2)
      GO TO 9998
      OPEN(UNIT=24,DEVICE='DSK')
      DO  12   K=1,NLS+2
      READ(24,537)((AS(K,I,J),I=1,NMAX),J=1,MMAX)
      READ(24,537)((BS(K,I,J),I=1,NMAX),J=1,MMAX)
      READ(24,537)((CV(K,I,J),I=1,NMAX),J=1,MMAX)
      READ(24,537)((DV(K,I,J),I=1,NMAX),J=1,MMAX)
12    CONTINUE
537   FORMAT(10E13.7)
9998  CONTINUE
      CALL RTIME(ISEG3)
C-------------------------------------------------------------
C
C     calculate the collocation matrix
C
      DO  13   I=1,NC
      DO  14   K=1,NLS+2
      IF(I.EQ.NLS+2)GO TO 101
      CC(I,K)=BS(K,I+NW,1)-YPC(I)*AS(K,I+NW,1)
      CC(I,K+NLS+2)=DV(K,I+NW,1)-YPC(I)*CV(K,I+NW,1)
      GO TO 102
101   CC(I,K)=AS(K,I+NW,1)
      CC(I,K+NLS+2)=CV(K,I+NW,1)
102   IF(I.EQ.NC)CC(I+1,K)=0.0
      IF(I.EQ.NC)CC(I+1,K+NLS+2)=0.0
      IF(I.EQ.NC)CC(I+1,NLS+3)=1.0
14    CONTINUE
13    CONTINUE
C-------------------------------------------------------------
C
C     invert the matrix CC(I,K)
C
      NN=NC+1;   IDGT=0;   IA=50
      CALL F01AAF(CC,IA,NN,CCINV,IA,WKAREA,IDGT)
C-------------------------------------------------------------
C
      FLAG=.FALSE.
C
C     initialise
C
      DO  15   I=1,NMAX
      FFC(I)=0.0;   FFCP(I)=0.0
      DO  16   J=1,MMAX
      UFNL(I,J)=0.0;   UFNL(I,J)=0.0
      GP(I,J)=0.0;   UF(I,J)=0.0;   VF(I,J)=0.0
16    CONTINUE
15    CONTINUE
C-------------------------------------------------------------
C
C     start iteration
C
      IT=0
103   CONTINUE
      DO  17   I=1,NC
      IF(I.EQ.NLS+2)GO TO 104
      ZETA(I)=CSALFA*YPC(I)-SNALFA-FFC(I)
      GO TO 17
```

```fortran
104       ZETA(I)=-CSALFA-VFC(I)
17        CONTINUE
          ZETA(JC+1)=0.0
          DO 18  I=1,JC+1
          SUM=0.0
          DO 19  K=1,JC+1
          SUM=SUM+COTNV(I,K)*ZETA(K)
19        CONTINUE
          IF(I.LE.NUS+2)SIG(I)=SUM
          IF(I.GE.NUS+2)GAMA(I-NUS-2)=SUM
18        CONTINUE
C
C         compute the total velocity field
C
          CALL UVPSIG
C
          IF(IT.GT.0)GO TO 105
C         -----------------------------------------------------------------
C
C         Incompressible results
C
          DO 20  I=1,NC
          UCOMP=CSALFA+UF(NN+I,1);   UCOMP2=UCOMP**2
          VCOMP=SNALFA+VF(NN+I,1);   VCOMP2=VCOMP**2
          IF(I.LE.NUS+2)CPLINU(I)=1.-UCOMP2-VCOMP2
          IF(I.GE.NUS+2)CPLINL(I-NUS-1)=1.-UCOMP2-VCOMP2
          CPLIN(I)=1.-UCOMP2-VCOMP2
20        CONTINUE
          ITEU=1;    ITEL=NC;   XM=0.0
          CALL FORCE(ITEU,ITEL,XC,YC,CPLIN,ALFA,XM,CL,CD,CM)
C         -----------------------------------------------------------------
          WRITE(20,510)
          WRITE(20,511)CL,CD,XM,CM
          WRITE(20,513)
          DO 31  I=1,NC
          WRITE(20,512)XC(I),CPLIN(I)
31        CONTINUE
          WRITE(20,701)
C         -----------------------------------------------------------------
C
C         calculate the field sources
C
105       CONTINUE
          CALL DERIV(UF,UFX,UFY)
          DO 21  J=1,MMAX
          DO 22  I=1,NMAX
          VFX(I,J)=UFY(I,J)
          VFY(I,J)=GF(I,J)-UFX(I,J)
          UCOMP=CSALFA+UF(I,J);   VCOMP=SNALFA+VF(I,J)
          Q(I,J)=UCOMP**2+VCOMP**2
          A2(I,J)=1./FSM2+0.2*(1.-Q(I,J))
          RMACH(I,J)=SQRT(Q(I,J)/A2(I,J))
          Q2(I,J)=0.5*Q(I,J)
          Q2X(I,J)=UCOMP*UFX(I,J)+VCOMP*VFX(I,J)
          Q2Y(I,J)=UCOMP*UFY(I,J)+VCOMP*VFY(I,J)
          QS(I,J)=(UCOMP*Q2X(I,J)+VCOMP*Q2Y(I,J))/Q(I,J)
          STERM1=ALP1*(UF(I,J)-UFP(I,J))
          STERM2=ALP2*(VF(I,J)-VFP(I,J))
          GFN(I,J)=Q(I,J)*QS(I,J)/A2(I,J)+STERM1+STERM2
          RHO(I,J)=(1.-FGAM*(Q(I,J)-1.0))**EXPGAM
          RMU(I,J)=AQ-BQ/Q(I,J)
          IF(RMU(I,J).LT.0.000001)RMU(I,J)=0.0
          Q(I,J)=SQRT(Q(I,J))
          IF(FLAG)GO TO 22
          IF(Q(I,J).LT.QSONIC)GO TO 22
          WRITE(6,515);  WRITE(20,515);       FLAG=.TRUE.
515       FORMAT(/15X,'flow is supercritical'/)
22        CONTINUE
21        CONTINUE
          GO TO 9997
          CALL DERIV(RHO,RHOX,RHOY)
          DO 50  J=1,MMAX
          DO 50  I=1,NMAX
          UCOMP=CSALFA+UF(I,J);       VCOMP=SNALFA+VF(I,J)
          GFN(I,J)=-(UCOMP*RHOX(I,J)+VCOMP*RHOY(I,J))/RHO(I,J)
```

```
50      CONTINUE
9997    CONTINUE
        CALL APPVFLS
        IF(RNORM.AVRG)GO TO 109
        DO 51   I=1,NMAX
        GFN(I,1)=GFN(I,1)+0.25*(GFN(I,2)-GFN(I,1))
51      CONTINUE
109     CONTINUE
108     CONTINUE
        RNORM=0.0
        DO 25   I=1,NMAX
        DO 25   J=1,NMAX
        R(I,J)=ABS(GFN(I,J)-GF(I,J))
        IF(R(I,J).GT.RNORM)RNORM=R(I,J)
        GF(I,J)=GFN(I,J)
26      CONTINUE
25      CONTINUE
        DO 53   I=1,NMAX
        CP(I)=PRF*((1.-FGAM*(Q(I,1)**2-1.0))**(GAM*EXPGAM)-1.0)
53      CONTINUE
        IF(MOD(IT,1).EQ.0)WRITE(20,555)IT,(CP(I),I=1,NMAX)
        WRITE(20,520)IT,RNORM;   WRITE(5,520)IT,RNORM
520     FORMAT(15X,'IT=',I3,3X,'RNORM=',F15.7)
        IF(RNORM.LT.EPS)GO TO 1001;   IF(RNORM.GT.50.0)GO TO 1002
        IT=IT+1;   IF(IT.GT.ITMAX)GO TO 106;   GO TO 107
C
106     WRITE(5,521)
        READ(5,*)ITMAX;   IF(ITMAX.EQ.0)GO TO 1002
C
107     CONTINUE
C
C       Compute the induced velocities due to the field sources and
C       then calculate the non-linear forcing function vector FFC
C
        CALL UVFGF2
        CALL FFCCAL
C
C       FFCCAL calculates the forcing function at the collocation points
C       IF(IT.EQ.1)CALL RTIME(TSEG4)
        GO TO 103
C
C       basic iterative segment ends
C
C
1002    WRITE(20,522)IT,RNORM;       GO TO 1003
1001    WRITE(20,523)IT,RNORM
1003    CONTINUE
        DO 27   I=1,NMAX
        CP(I)=PRF*((1.-FGAM*(Q(I,1)**2-1.0))**(GAM*EXPGAM)-1.0)
27      CONTINUE
        DO 28   I=1,NMAX
        XSU(I)=REAL(ZGRID(I,1))
        YSU(I)=AIMAG(ZGRID(I,1))
28      CONTINUE
        DO 29   I=1,NC
        CPCH(I)=CP(I+NW)
        XCH(I)=REAL(ZGRID(I+NW,1))
        YCH(I)=AIMAG(ZGRID(I+NW,1))
29      CONTINUE
        ITEU=1;     ITEL=NC;     XM=0.0
        CALL FORCE(ITEU,ITEL,XCH,YCH,CPCH,ALFA,XM,CL,CD,CM)
        CALL RTIME(ISEG5);   ITOTAL=ISEG5-ISEG1
        WRITE(20,701)
        WRITE(20,525)
        WRITE(20,541)CL,CD,XM,CM
        WRITE(20,526)
526     FORMAT(/20X,'XSU(I)',7X,'CP(I)',9X,'Q(I,1)',7X,'RMACH(I,1)')
525     FORMAT(/30X,'COMPRESSIBLE FLOW RESULTS'//)
        DO 30   I=1,NMAX
        WRITE(20,524)XSU(I),CP(I),Q(I,1),RMACH(I,1)
30      CONTINUE
        WRITE(20,531)ITOTAL
        WRITE(20,501)
C
C       program continuation for further calculations at other Mach
```

```
C     numbers/angle of attack/viscosity parameter.
C
C     the iteration parameters ALP1,ALP2 may also be varied depending
C     upon the performance of the iterative scheme so far.
C
      WRITE(5,558)
      READ(5,*)ICAL
      IF(ICAL.EQ.0)STOP
      WRITE(5,559)FSM,ALFAD
      READ(5,*)FSM,ALFA
      WRITE(5,561)ALP1,ALP2,LAMAX
      READ(5,*)ALP1,ALP2,LAMAX
C     -----------------------------------------------------------------
      CALL RTIME(ISEG1)
      PIR=3.141592653589793; EXPGAM=1./(GAM-1.); PIF=0.5/PIR
      EXPGAM=1./(GAM-1.); FSM2=FSM**2; FGAM=0.5*(GAM-1.)*FSM2
      PRF=2./GAM/FSM2; AQ=0.5*(1.+GAM); BQ=1./FSM2+0.5*(GAM-1.)
      QSONIC=SQRT(BQ/AQ); ALFAD=ALFA; TALFA=ALFA; GAM1=1.+GAM
      CPSTAR=PRF*((1.-FGAM*(QSONIC**2-1.))**(GAM*EXPGAM)-1.0)
      ALFA=ALFA*PIR/180.; CSALFA=COS(ALFA); SNALFA=SIN(ALFA)
      WRITE(20,560)
      WRITE(20,503)FSM,TAU,ALFAD
      WRITE(20,509)QSONIC,CPSTAR
      WRITE(20,562)ALP1,ALP2,LAMAX
      WRITE(20,501)
C     -----------------------------------------------------------------
C     ITMAX=5;  IT=1;        GO TO 107
C     -----------------------------------------------------------------
C
  501 FORMAT(2X,120(1H-))
  524 FORMAT(15X,4(F10.4,4X))
  513 FORMAT(20X,'XC(I)',8X,'CPLTN(I)')
  512 FORMAT(15X,F10.4,5X,F10.4)
  503 FORMAT(/15X,'free stream mach number FSM=',F5.3/15X,'airfoil
     1 thickness TAU=',F5.3/15X,'angle of attack ALFAD=',F5.2/
     115X,'artificial viscosity factor LAMAX=',F6.3)
  504 FORMAT(/15X,'number of line  elements NLS=',I3/15X,
     1'number of collocation points NC=',I3)
  505 FORMAT(/15X,'GRID DETAILS:',2X,'NMAX=',I3,8X,'MMAX=',I3,10X'NW='
     1I3/30X,'XMAX=',F6.2,5X,'XMIN=',F6.2,5X,'YMAX=',F6.2)
  507 FORMAT(15X,'number of points before expected fore shock location
     1 NBLC=',I2/15X,'after downstream shock location NALC=',I2/
     115X,'number of points in the expected shock interval NILC=',
     1I2/15X,'shock interval XSHL< X < XSHR  XSHL=',F6.3,2X,'XSHR=',
     1F6.3)
  509 FORMAT(/15X,'sonic velocity QSONIC=',F7.3/15X,'sonic pressure
     1 coefficient CPSTAR=',F7.3)
  521 FORMAT(2X,'ITMAX limit exceeded , type new ITMAX or type zero
     1 to stop')
  522 FORMAT(15X,'unconverged results IT=',I3,5X,'RNORM=',F10.4)
  523 FORMAT(15X,'converged results IT=',I3,5X,'RNORM=',F10.4)
  531 FORMAT(/20X,'total execution time ITOTAL=',I7,'millisecs')
  540 FORMAT(/30X,'INCOMPRESSIBLE FLOW RESULTS'//)
  541 FORMAT(/15X,'lift coeff CL=',F10.5/15X,'drag coeff CD=',
     1F10.5/15X,'moment coef about XM=',F5.2,2X,'is CM=',F10.6/)
  557 FORMAT(/15X,'airfoil specification:',A15/15X,'viscosity
     1scheme:'A10)
  558 FORMAT(/15X,'want to carry out calculations for other'/
     115X,'Mach number or angle of attack?'/15X,'type 0 for NO and
     1any other number for YES')
  559 FORMAT(/15X,'type new Mach number and angle of attack '/
     115X,'present values are: FSM=',F6.3,2X,'ALFA=',F6.3)
  560 FORMAT(/15X,'calculations continued for higher Mach number/
     1angle of attack'/)
  561 FORMAT(/15X,'type values for ALP1,ALP2,LAMAX'/15X,'present value
     1 are'2X,3(F6.2,2X),'respectively')
  562 FORMAT(/15X,'new values for ALP1,ALP2,LAMAX are'/15X,3(F6.3,2X),
     1'respectively')
   99 FORMAT(2A6)
  565 FORMAT(15X,'calculations carried out with GF(I,J) averaged'/
     115X,'for the first row of field panels')
  566 FORMAT(15X,'calculations without GF(I,J) averaging'/
     115X,'for the first row of field panels')
  555 FORMAT(15X,'nonlinear pressure distribution at IT=',I3/(15X,5(
     1F10.4,2X))/)
      STOP
```

```fortran
      END
C     ***********************************************************************
      SUBROUTINE UVFSIG
C
C     This subroutine calculates the induced velocities due to the
C     internal singularities ,adds them to the non-linear contri-
C     butions and returns the total perturbation velocities
C
      COMMON /F1/ NLS,NMAX,MMAX,NW,NC,TCAM,IALFA
      COMMON /C3/ AS(23,51,5),BS(23,51,5),CV(23,51,5),DV(23,51,5)
      COMMON /C4/ UFP(51,5),VFP(51,5)
      COMMON /C10/ UF(51,5),VF(51,5)
      COMMON /C13/ UFNL(51,5),VFNL(51,5)
      COMMON /C15/ SIG(55),GAMA(55)
      DO 10  I=1,MMAX
      DO 11  I=1,NMAX
      SUMU=0.0;  SUMV=0.0
      DO 12  K=1,NLS+2
      SUMU=SUMU+AS(K,I,J)*SIG(K)+CV(K,I,J)*GAMA(K)
      SUMV=SUMV+BS(K,I,J)*SIG(K)+DV(K,I,J)*GAMA(K)
12    CONTINUE
      UFP(I,J)=UF(I,J);    VFP(I,J)=VF(I,J)
      UF(I,J)=SUMU+UFNL(I,J);  VF(I,J)=SUMV+VFNL(I,J)
11    CONTINUE
10    CONTINUE
      RETURN
      END
C     ***********************************************************************
      SUBROUTINE UVFGF2
C
C     This subroutine calculates the non-linear contribution to
C     the perturbation velocity components.
C
      COMMON /F1/ NLS,NMAX,MMAX,NW,NC,TCAM,IALFA
      COMMON /C5/ FSV(51,26,5),FSU(51,5,26,5)
      COMMON /C13/ UFNL(51,5),VFNL(51,5)
      COMMON /C14/ GFN(51,5),R(51,5),GF(51,5)
      IMAX=NMAX;  KMAX=NMAX
      IF(ICAM.EQ.0)KMAX=(NMAX+1)/2
      DO 10  J=1,MMAX
      DO 11  I=1,IMAX
      ICOMP=NMAX-I+1
      SUMU=0.0;  SUMV=0.0
      DO 12  L=1,MMAX
      DO 13  K=1,KMAX
      IF(ICAM.NE.0)GO TO 102
      KCOMP=NMAX-K+1
      IF(K.EQ.KMAX)GO TO 101
      SUMU=SUMU+FSU(I,J,K,L)*GF(K,L)+FSU(ICOMP,J,K,L)*GF(KCOMP,L)
      IF(J.EQ.1)SUMV=SUMV+FSV(I,K,L)*GF(K,L)-FSV(ICOMP,K,L)*
     1GF(KCOMP,L)
      GO TO 13
101   SUMU=SUMU+FSU(I,J,K,L)*GF(K,L)
      IF(J.EQ.1)SUMV=SUMV+FSV(I,K,L)*GF(K,L)
      GO TO 13
102   SUMU=SUMU+FSU(I,J,K,L)*GF(K,L)
      IF(J.EQ.1)SUMV=SUMV+FSV(I,K,L)*GF(K,L)
13    CONTINUE
12    CONTINUE
      UFNL(I,J)=SUMU
      IF(J.EQ.1)VFNL(I,J)=SUMV
11    CONTINUE
10    CONTINUE
506   FORMAT(10F13.8)
      RETURN
      END
C     ***********************************************************************
      SUBROUTINE FFCCAL
C
C     This subroutine calculates the forcing function due to the
C     non-linear term in the implementation of the tangency
C     boundary condition on the body surface.
C
      COMMON /F1/ NLS,NMAX,MMAX,NW,NC,TCAM,IALFA
      COMMON /C2/ XC(55),YC(55),YPC(55),DSP(55)
```

```
      COMMON /C7/ FFC(55),FFCP(55)
      COMMON /C13/ UFNL(51,5),VFNL(51,5)
      DO  10  I=1,NC
      IF(I.EO.NLS+2)GO TO 101
      FFC(I)=VFNL(I+NW,1)-YPC(I)*UFNL(I+NW,1);  GO TO 10
101   FFC(I)=UFNL(I+NW,1)
10    CONTINUE
      FFC(NC+1)=0.0
      RETURN
      END
C     **************************************************************
      SUBROUTINE DERIV(Q,QX,QY)
C
C     This subroutine calculates derivatives on the transformed
C     plane and computes the physical plane derivatives using
C     the JACOBIAN of the transformation.
C
      DIMENSION Q(51,5),QX(51,5),QY(51,5)
      COMMON /E1/ NLS,NMAX,MMAX,NW,NC,TCAM,IALFA
      COMMON /C16/ DJACOB(51,5,2,2)
      COMMON /C17/ A(51,3),B(10,3)
      DO  10  J=1,MMAX
      DO  11  I=1,NMAX
      IF(I.EO.1)GO TO 105
      IF(I.EO.NMAX)GO TO 106
      IF(I.EO.NW+1)GO TO 105
      IF(NW.EO.0)GO TO 101
      IF(I.EO.NW)GO TO 106
101   CONTINUE
      IF(I.EO.NMAX-NW)GO TO 106
      IF(I.EO.NMAX-NW+1)GO TO 105
      QX(I,J)=A(I,1)*Q(I+1,J)+A(I,2)*Q(I,J)+A(I,3)*Q(I-1,J);  GO TO 11
105   QX(I,J)=A(I,1)*Q(I+2,J)+A(I,2)*Q(I+1,J)+A(I,3)*Q(I,J);  GO TO 11
106   QX(I,J)=A(I,1)*Q(I,J)+A(I,2)*Q(I-1,J)+A(I,3)*Q(I-2,J)
11    CONTINUE
10    CONTINUE
      DO  12  I=1,MMAX
      DO  13  J=1,NMAX
      IF(J.EO.1)GO TO 107
      IF(J.EO.MMAX)GO TO 108
      QY(I,J)=B(J,1)*Q(I,J+1)+B(J,2)*Q(I,J)+B(J,3)*Q(I,J-1);  GO TO 13
107   QY(I,J)=B(J,1)*Q(I,J+2)+B(J,2)*Q(I,J+1)+B(J,3)*Q(I,J);  GO TO 13
108   QY(I,J)=B(J,1)*Q(I,J)+B(J,2)*Q(I,J-1)+B(J,3)*Q(I,J-2)
13    CONTINUE
12    CONTINUE
C     transform to (X,Y) plane
      DO  14  J=1,MMAX
      DO  15  I=1,NMAX
      SUM1=DJACOB(I,J,1,1)*QX(I,J)+DJACOB(I,J,1,2)*QY(I,J)
      SUM2=DJACOB(I,J,2,1)*QX(I,J)+DJACOB(I,J,2,2)*QY(I,J)
      QX(I,J)=SUM1;  QY(I,J)=SUM2
15    CONTINUE
14    CONTINUE
      RETURN
      END
C     **************************************************************
      SUBROUTINE DERCOF
C
C     This subroutine computes the coefficients for derivative
C     calculations in DERIV.
C
      COMMON /E1/ NLS,NMAX,MMAX,NW,NC,TCAM,IALFA
      COMMON /C18/ DXI(55),DBET(10)
      COMMON /C17/ A(51,3),B(10,3)
      DO  16  I=1,NMAX
      IF(I.EO.1)GO TO 101
      IF(I.EO.NW+1)GO TO 101
      IF(NW.EO.0)GO TO 105
      IF(I.EO.NW)GO TO 102
105   CONTINUE
      IF(I.EO.NMAX)GO TO 102
      IF(I.EO.NMAX-NW)GO TO 102
      IF(I.EO.NMAX-NW+1)GO TO 101
      SUM=DXI(I-1)+DXI(I)
      PROD=DXI(I-1)*DXI(I)
```

```
      A(I,1)=DXI(I-1)/DXI(I)/SUM
      A(I,2)=(DXI(I)-DXI(I-1))/PROD
      A(I,3)=-DXI(I)/DXI(I-1)/SUM
      GO TO 16
101   SUM=DXI(I)+DXI(I+1)
      PROD=DXI(I)*DXI(I+1)
      A(I,1)=-DXI(I)/DXI(I+1)/SUM
      A(I,2)=SUM/PROD
      A(I,3)=-(SUM+DXI(I))/DXI(I)/SUM
      GO TO 16
102   SUM=DXI(I-1)+DXI(I-2)
      PROD=DXI(I-1)*DXI(I-2)
      A(I,1)=(SUM+DXI(I-1))/DXI(I-1)/SUM
      A(I,2)=-SUM/PROD
      A(I,3)=DXI(I-1)/DXI(I-2)/SUM
16    CONTINUE
      DO  17  I=1,MMAX
      IF(I.EQ.1)GO TO 103
      IF(I.EQ.MMAX)GO TO 104
      SUM=DBET(I-1)+DBET(I)
      PROD=DBET(I-1)*DBET(I)
      B(I,1)=DBET(I-1)/DBET(I)/SUM
      B(I,2)=(DBET(I)-DBET(I-1))/PROD
      B(I,3)=-DBET(I)/DBET(I-1)/SUM
      GO TO 17
103   SUM=DBET(I)+DBET(I+1)
      PROD=DBET(I)*DBET(I+1)
      B(I,1)=-DBET(I)/DBET(I+1)/SUM
      B(I,2)=SUM/PROD
      B(I,3)=-(SUM+DBET(I))/DBET(I)/SUM
      GO TO 17
104   SUM=DBET(I-1)+DBET(I-2)
      PROD=DBET(I-1)*DBET(I-2)
      B(I,1)=(SUM+DBET(I-1))/DBET(I-1)/SUM
      B(I,2)=-SUM/PROD
      B(I,3)=DBET(I-1)/DBET(I-2)/SUM
17    CONTINUE
      RETURN
      END
C     ***********************************************************
C     ***********************************************************
      SUBROUTINE FORCE(I1,I2,X,Y,CP,ALPHA,XM,CL,CD,CM)
C
C     computes lift drag and moment coefficients by trapezoidal rule
C
      DIMENSION X(55),Y(55),CP(55)
      CL     =0.0;      CD=0.0;          CM=0.0;          N=I2-1
      DO   12   I=I1,N
      DX     =X(I+1)-X(I);               DY        =Y(I+1)-Y(I)
      XA     =0.5*(X(I+1)+X(I))-XM;      YA        =0.5*(Y(I+1)+Y(I))
      CPA    =0.5*(CP(I+1)+CP(I));       DCL       =-CPA*DX
      DCD    =CPA*DY;         CL=CL+DCL; CD        =CD+DCD
12    CM     =CM+DCD*YA-DCL*XA
C
C     rotate CL and CD to direction of free stream
C
      DCL=CL*COS(ALPHA)-CD*SIN(ALPHA)
      CD =CL*SIN(ALPHA)+CD*COS(ALPHA)
      CL =DCL
      CL=-CL     ;          CM=-CM
      RETURN
      END
C     ***********************************************************
```

```
C     THIS IS FILE GRID.FOR

      SUBROUTINE GRID(NPLS,NTLS,NALS,NI,NJ,IC,SEJ,NJLAST,BETMIN,ORIGIN
     1,DELTA,BPSIE,XSHL,XSHR,ICAM,VMAX,AMAX,NLS,NC,DELTA,CHORD)

C     This subroutine computes the grid co-ordinates and the JACOBIAN
C     of the transformation.

      IMPLICIT COMPLEX(Z,W)
      DIMENSION BETAYI(55),BETAYIT(55)
      DIMENSION Z(55),W(55)
      DIMENSION WGRID(52,6),AGRID(52,6)
      DIMENSION XS(55),XI(55),BETAJ(10),BETAM(10),BIAXTN(55)
      COMMON /B1/ RADIE,TAVI1,IAFCR2,TAU,PI(10),CT(10)
      COMMON /C1/ XS(55),YS(55),YPS(55),DS(55),SINP(55),COSI(55)
      COMMON /C2/ XC(55),YC(55),YPC(55),DSP(55)
      COMMON /C5/ ZGRID(52,6),ZWPI(52,6)
      COMMON /C10/ DXI(55),DSET(10)
      COMMON /C15/ DJACOB(51,5,2,2)
      COMMON /C21/ DELXIT(51,5)
      ZETA(Z)=1.-4.*CEXP(Z)
      PIR=3.14159265358979323;   ZPI=CMPLX(0.,2.*PIR)
      NLS=NALS+NTLS+NALS
      NJT=NJ+1;   NXI=2*(NLS+NA)+3
      VMAX=NYL;   AMAX=NJ-NJLAST;   NC=2*NLS+3
      XS(1)=1.0;   XS(2)=1.0-BPSIE
      DXLS=(1.0-BPSIE-XSHR)/FLOAT(NALS-1)
      DO 11 I=3,NALS+1
      XS(I)=XS(I-1)-DXLS
   11 CONTINUE
      NST=NALS+NTLS+1
      DXLS=(XSHR-XSHL)/FLOAT(NTLS)
      DO 12 I=NALS+2,NST
      XS(I)=XS(I-1)-DXLS
   12 CONTINUE
      RINT=XSHL-DELTA
      DO 37 I=NST+1,NLS+1
      THETA=0.5*PIR*FLOAT(I-NST)/FLOAT(NLS-NST+1)
      XX=RINT*SIN(THETA)
      XS(I)=XSHL-XX
   37 CONTINUE
      DO 13 I=1,NLS
      XC(I)=0.5*(XS(I)+XS(I+1))
   13 CONTINUE
      XC(NLS+1)=0.5*XS(NLS+1)
      XC(NLS+2)=0.0
      XC(NLS+3)=XC(NLS+1)
      DO 14 I=NLS+4,NC
      XC(I)=XC(NC-I+1)
   14 CONTINUE
      CALL AERFOL(NLS,NC,ICAM,DELTA,ORIGIN,CHORD)
      DO 16 I=1,NLS+1
      Z(I)=CMPLX(XC(I)-ORIGIN,YC(I))
      IF(ICAM.NE.0)Z(NC-I+1)=CMPLX(XC(NC-I+1)-ORIGIN,YC(NC-I+1))
      IF(ICAM.EQ.0)Z(NC-I+1)=CONJG(Z(I))
   16 CONTINUE
      Z(NLS+2)=CMPLX(-ORIGIN,0.0)
      DO 17 I=1,NC
      ZETA1=ZETA(Z(I))
      ZETA2=CSQRT(ZETA1**2-1.0)
      W(I)=CLOG(ZETA1+ZETA2)
   17 CONTINUE
C     ------------------------------------------------------------
C     Inverse transformation
C     ------------------------------------------------------------
      DO 18 I=1,NC
      WCOSH=(CEXP(W(I))+CEXP(-W(I)))/2.
      ZARG=(1.0-WCOSH)/2.
      ZINV=CLOG(ZARG)
      IF(REAL(W(I)).GT.0.0)ZINV=ZINV-ZPI
      ZERROR=Z(I)-ZINV
      IF(CABS(ZERROR).LT.0.00001)GO TO 18
  708 FORMAT(2X,'error in inverse transformation  program is
```

```
     1 terminated, the current values of I,Z(I),ZTAV are'/2X,'I=',
    102,2X,2(E10.5,',',E10.5,3X))
18   CONTINUE
C    ------------------------------------------------------------
C    calculation of grid points in (xi,eta) plane and trans-
C    -formation to (x,y) plane.
C    ------------------------------------------------------------
     XITE=1.0;   YITE=-.80005
     ZITE=CMPLX(YITE-DRTSIN,ITE)
     ZETA1=ZITE/A(ZTE);   ZETA2=CSQRT(ZETA1**2-1.0)
     ZIT=CLOG(ZETA1+ZETA2)
     XITEL=REAL(ZITE);    YITEP=-XITEL
     ETATEL=AIMAG(ZITE);  ETATER=ETATEL
     DXITE=ABS(REAL(W(1))-XITEL);  DXITE1=10.*DXITE
     IF(NW.EQ.0)GO TO 101
     XIG(NW)=YITEL-DXITE
     ETAXIG(NW)=ETATEL
     DXINC=DXITE
     DO  19  I=1,NW-1
     XIG(NW-I)=XIG(NW-I+1)-DXINC
     ETAXIG(NW-I)=ETATEL
     DXINC=2.*DXINC
19   CONTINUE
101  CONTINUE
     DO  20  I=1,NC
     XIG(NW+I)=REAL(W(I))
     ETAXIG(NW+I)=AIMAG(W(I))
20   CONTINUE
     IF(NW.EQ.0)GO TO 102
     DO  21  I=1,NW
     XIG(NW+NC+I)=-XIG(NW-I+1)
     ETAXIG(NW+NC+I)=ETAXIG(NW-I+1)
21   CONTINUE
102  CONTINUE
     DO  25  I=1,NXI-1
     DXI(I)=XIG(I+1)-XIG(I)
25   CONTINUE
     DXINC=0.5*(XTG(NLS+2+NW)-XTG(NLS+1+NW))
     XIN(NLS+2+NW)=XIG(NLS+2+NW)+DXINC
     ETAXIN(NLS+3+NW)=0.5*(ETAXIG(NLS+3+NW)+ETAXIG(NLS+2+NW))
     DO  34  I=1,NLS
     XIN(NLS+3+NW-I)=0.5*(XTG(NLS+3+NW-I)+XTG(NLS+2+NW-I))
     ETAXIN(NLS+3+NW-I)=0.5*(ETAXIG(NLS+3+NW-I)+ETAXIG(NLS+2+NW-I))
34   CONTINUE
     XIN(NW+1)=XITEL;   ETAXIN(NW+1)=ETATEL
     XIN(NW+2)=XIG(NW+1)+DXITE
     ETAXIN(NW+2)=ETATEL-2.*(ETATEL-ETAXIG(NW+1))
     IF(NW.EQ.0)GO TO 103
     XIN(NW)=XIG(NW)-DXITE;   ETAXIN(NW)=ETATEL
     DXINC=XIN(NW)-XIG(NW-1)
     DO  26  I=1,NW-1
     XIN(NW-I)=XIG(NW-I)-DXINC
     ETAXIN(NW-I)=ETATEL
     IF(I.LE.NW-2)DXINC=XIN(NW-I)-XIG(NW-I-1)
26   CONTINUE
103  CONTINUE
     DO  35  I=NW+NLS+4,NXI-NW-1
     XIN(I)=0.5*(XIG(I-1)+XIG(I))
     ETAXIN(I)=0.5*(ETAXIG(I-1)+ETAXIG(I))
35   CONTINUE
     XIN(NXI-NW)=XIG(NXI-NW)-DXITE
     ETAXIN(NXI-NW)=ETATER-2.*(ETATER-ETAXIG(NXI-NW))
     XIN(NXI-NW+1)=XITER
     ETAXIN(NXI-NW+1)=ETATER
     IF(NW.EQ.0)GO TO 104
     DO  36  I=NXI-NW+2,NXI+1
     XIN(I)=-XIN(NXI+2-I)
     ETAXIN(I)=ETAXIN(NXI+2-I)
36   CONTINUE
104  CONTINUE
     DO  27  I=1,NXI
     DETAXI(I)=(ETAXIN(I+1)-ETAXIN(I))/(XIN(I+1)-XIN(I))
27   CONTINUE
     DELBET=(1.-BETMIN)/(SFJ*NJ+JC*(1.-SFJ)-0.5)
     BETAN(I)=1.0
```

```
      DO 22    J=2,JC+1
      BETAN(J)=BETAN(J-1)+DELBET
      IF(J.EQ.0)BETAN(J)=BETAN(J-1)-0.5*DELBET
22    CONTINUE
      DELBET=SPI*DELBET
      DO   39    J=JC+2,JJ
      BETAN(J)=BETAN(J-1)+DELBET
39    CONTINUE
      DO   23    J=1,NJ
      BETAG(J)=0.5*(BETAN(J)+BETAN(J+1))
      IF(J.EQ.1)BETAG(J)=BETAN(J)
23    CONTINUE
      DO   24    J=1,NJ-1
      DBET(J)=BETAG(J+1)-BETAG(J)
24    CONTINUE
      DO   28    I=1,NAI+1
      XIG(I)=XIG(I);   DXIXIG=ETAXIG(I)
      XIN(I)=ATIF(I);  DIXIN=ETAXIJ(I)
      DO   29    J=1,NJ1
      ETANI=BETAN(I)*DBETAT
      WIN=CMPLX(XIN,ETANI);  WNET(I,J)=WIN
      WCOSH=(CEXP(WIN)+CEXP(-WIN))/2.
      ZARG=(1.-WCOSH)/2.
      ZZ=CLOG(ZARG)
      IF(REAL(WIN).GT.0.0)ZZ=ZZ-ZPI
      ZNET(I,J)=ZZ+ORIGIN
      IF(J.EQ.1.AND.I.GE.NW+NC+1)ZNET(I,J)=CONJG(ZNET(2*NW+NC+2-I,J))
      IF(I.EQ.NXI+1)GO TO 29
      IF(J.EQ.NJ1)GO TO 29
      ETAGJ=BETAG(I)*DETXIGT
      WIG=CMPLX(XIGI,ETAGJ)
      WGRID(I,J)=WIG
      WCOSH=(CEXP(WIG)+CEXP(-WIG))/2.
      ZARG=(1.-WCOSH)/2.;   ZZ=CLOG(ZARG)
      IF(REAL(WIG).GT.0.0)ZZ=ZZ-ZPI
      ZGRID(I,J)=ZZ+ORIGIN
      IF(J.EQ.1.AND.I.GT.NW+NC)ZGRID(I,J)=CONJG(ZGRID(2*NW+NC+1-I,J))
29    CONTINUE
28    CONTINUE
      DO   40    I=1,NMAX-1
      X1=REAL(ZGRID(I,1))-REAL(ZGRID(I+1,1))
      Y1=AIMAG(ZGRID(I,1))-AIMAG(ZGRID(I+1,1))
      IF(I.LE.NLS+2+NJ)DSP(I)=-SQRT(X1**2+Y1**2)
      IF(I.GT.NLS+2+NW)DSP(I)=SQRT(X1**2+Y1**2)
40    CONTINUE
      IMID=(NMAX+1)/2
      DO   41    J=1,NMAX
      DO   41    I=1,NMAX
      IF(I.LT.IMID)DELXIJ(I,J)=-REAL(ZGRID(I+1,J)-ZGRID(I,J))
      IF(I.GE.IMID)DELXIJ(I,J)=REAL(ZGRID(I+1,J)-ZGRID(I,J))
41    CONTINUE
CCC -----------------------------------------------------------------
C     calculate the JACOBIAN
CCC -----------------------------------------------------------------
      DO   30    J=1,NJ
      DO   31    I=1,NMAX
      ZGP=ZGRID(I,J)+DELX-ORIGIN
      ZGM=ZGRID(I,J)-DELX-ORIGIN
      ZETAP1=ZETA(ZGP);   ZETAM1=ZETA(ZGM)
      ZETAP2=CSQRT(ZETAP1**2-1.)
      ZETAM2=CSQRT(ZETAM1**2-1.)
      WIP=CLOG(ZETAP1+ZETAP2)
      WIM=CLOG(ZETAM1+ZETAM2)
      DXIDX=0.5*(REAL(WIP)-REAL(WIM))/DELX
      DETADX=0.5*(AIMAG(WIP)-AIMAG(WIM))/DELX
      DJACOB(I,J,1,1)=DXIDX
      DJACOB(I,J,2,1)=-DETADX
      FACTOR=BETAG(J)*DETAXI(I)/ETAXIG(I)
      D12=-FACTOR*DXIDX+DETADX/ETAXIG(I)
      D22=FACTOR*DETADX+DXIDX/ETAXIG(I)
      DJACOB(I,J,1,2)=D12
      DJACOB(I,J,2,2)=D22
31    CONTINUE
30    CONTINUE
      RETURN
9999  CONTINUE
      RETURN
      END
C     ***************************************************************
```

```
C
C
C
C        THIS IS FILE AERFOL.FOR
C
C        SUBROUTINE AERFOL(NLS,NC,ICAM,DELTA,ORIGIN,CHORD)
C
C        This subroutine accepts aerfoil data and calculates collocation
C        points and internal singularity locations by spline interpola-
C        tion. Liberal insertion of comments in the routine explain
C        the various features of this routine. However the routine
C        has been tested and verified to work only for the cases
C        when the airfoil is prescribed analytically or in terms of
C        upper and lower surface coordinates.
C
         DIMENSION XINU(150),YINU(150),YINU1(150),YINU2(150),YINU3(150)
         DIMENSION XINL(150),YINL(150),YINL1(150),YINL2(150),YINL3(150)
         DIMENSION YINMN(150),YINMN(150),YINMN1(150),YINMN2(150),
        1YINMN3(150)
         DIMENSION DUMMY(150),DUMMY1(150),DUMMY2(150)
         DIMENSION XLEU(3),YLEU(3),XLEL(3),YLEL(3)
         COMMON /E3/ RADLE,IAFCF1,IAFCF2,TAU,BT(10),CT(10)
         COMMON /C1/ XS(55),YS(55),YPS(55),DS(55),SINT(55),COST(55)
         COMMON /C2/ XC(55),YC(55),YPC(55),DSP(55)
         LOGICAL ANALTC
C        ------------------------------------------------------------------
         OPEN(UNIT=30,DEVICE='DSKC',FILE='AERFOL.OUT')
         OPEN(UNIT=1,DEVICE='DSKC',FILE='AFL.DAT')
         READ(1,*)ANALTC,IMIN
         READ(1,*)RADLE,TAU,ICAM,IDEFN
         ORIGIN=0.5*RADLE
C        ------------------------------------------------------------------
C
C        input data for profile prescribed analytically
C
         IF(ANALTC)READ(1,*)IAFCF1,(BT(I),I=1,IAFCF1)
         IF(ANALTC.AND.ICAM.NE.0)READ(1,*)IAFCF2,(CT(I),I=1,IAFCF2)
         IF(ANALTC.AND.IDEFN.EQ.0.AND.ICAM.NE.0)READ(1,*)NCORDU,
        1(DUMMY1(I),I=1,NCORDU)
         IF(ANALTC)GO TO 101
C        ------------------------------------------------------------------
C
C        input data for profile prescribed at discrete set of points
C
C
C        NOTE: the input profile coordinates must satisfy the following
C              conditions.
C              (i)   XINL(1)=XINU(1)      ;        YINL(1)=YINU(1)=0.0
C              (ii)  XINL(NCORDL)=XINU(NCORDU)
C
         READ(1,*)NCORDU,(XINU(I),I=1,NCORDU)
         READ(1,*)(YINU(I),I=1,NCORDU)
         IF(ICAM.EQ.0)GO TO 109
         READ(1,*)NCORDL,(XINL(I),I=1,NCORDL)
         READ(1,*)(YINL(I),I=1,NCORDL)
C        ------------------------------------------------------------------
C
C        calculation for non-analytic airfoil geometry
C
  109    CONTINUE
         CLOSE(UNIT=1,DEVICE='DSKC',FILE='AFL.DAT')
C        ------------------------------------------------------------------
         WRITE(30,527)
         WRITE(30,504)(XINU(I),I=1,NCORDU)
         WRITE(30,505)(YINU(I),I=1,NCORDU)
         WRITE(30,514)(XINL(I),I=1,NCORDL)
         WRITE(30,503)(YINL(I),I=1,NCORDL)
C        ------------------------------------------------------------------
C
C        check if the "input" profile conditions are satisfied
C
         IF(ICAM.EQ.0)GO TO 120
         IF(ABS(XINU(1)-XINL(1)).GT.0.0000001)WRITE(5,524)
         IF(ABS(XINU(1)-XINL(1)).GT.0.0000001)STOP
         IF(ABS(YINU(1)-YINL(1)).GT.0.0000001)WRITE(5,525)
```

```
      IF(ABS(YINU(1)-YINL(1)).GT.0.0000001)STOP
      IF(ABS(XINU(NCORDU)-XINL(NCORDL)).GT.0.0000001)WRITE(5,526)
      IF(ABS(XINU(NCORDU)-XINL(NCORDL)).GT.0.0000001)STOP
120   CONTINUE
C
C     check if the chord is normalised and the origin is at the
C     leading edge
C
      CHORD=XINU(NCORDU)-XINU(1)
      IF(CHORD-1.0.LT.0.0000001)GO TO 104
C
C     normalise the coordinates by the profile chord
C
      DO 14  I=1,NCORDU
      XINU(I)=XINU(I)/CHORD
      YINU(I)=YINU(I)/CHORD
14    CONTINUE
      IF(ICAM.EQ.0)GO TO 104
      DO 15  I=1,NCORDL
      XINL(I)=XINL(I)/CHORD
      YINL(I)=YINL(I)/CHORD
15    CONTINUE
C
104   CONTINUE
C
C     check if the origin is at the leading edge
C
      IF(ABS(XINU(1)).LT.0.0000001)GO TO 103
C
C     shift the origin to the leading edge
C
      SHIFT=XINU(1)
      DO 16  I=1,NCORDU
      XINU(I)=XINU(I)-SHIFT
16    CONTINUE
      IF(ICAM.EQ.0)GO TO 103
      SHIFT=XINL(1)
      DO 17  I=1,NCORDL
      XINL(I)=XINL(I)-SHIFT
17    CONTINUE
C
103   CONTINUE
C
C     determine input coordinates for FPLE to determine constants
C     appearing in approximate analytic expression
C
      DO 12  I=1,2
      XLEU(I)=XINU(I+IMIN)
      YLEU(I)=YINU(I+IMIN)
12    CONTINUE
      XLTU1=XLEU(1);    YLTU1=YLEU(1)
      IF(ICAM.EQ.0)GO TO 102
      DO 13  I=1,2
      XLEL(I)=XINL(I+IMIN)
      YLEL(I)=YINL(I+IMIN)
13    CONTINUE
      XLTL1=XLEL(1);    YLTL1=YLEL(1)
C
C     calculations for symmetric airfoil
C
102   CONTINUE
C
      IF(ICAM.NE.0)GO TO 112
C
C     calculations for symmetric airfoil
C
      DO 20  I=1,NLS
      YS(I)=0.0
      DS(I)=XS(I)-XS(I+1)
      SINT(I)=0.0;  COST(I)=1.0
20    CONTINUE
      M=1;  N=NCORDU;  IND=0;    XCI=XC(NLS+1)
      T1=FPLEU1(RADLE,XLTU1,YLTU1,XCI)
      T2=(YINU(N)-YINU(N-1))/(XINU(N)-XINU(N-1))
      CALL SPLTF(M,N,XINU,YINU,YINU1,YINU2,YINU3,1,T1,1,T2,0,0.0,IND)
```

```
              MI=1;    NI=NLS+2
              CALL INTPOL(MI,NT,XC,YC,YPC,M,N,YINU,YTNU,YINU1,YINU2,YINU3,0)
              DO   21   I=1,NLS+1
              YC(NC-I+1)=-YC(I)
              YPC(NC-I+1)=-YPC(I)
21            CONTINUE
              YC(NLS+2)=0.5;     YPC(NLS+2)=0.0
C
C             recompute YC(I),YPC(I) for XC(I) < 0.01
C
C             CALL FPLE(RADLE,XLEU,YLEU,XLEL,YLEL,TCAM,NLS,NC)
              GO TO 999
C             -------------------------------------------------------------------
112           CONTINUE
C
C             calculations for cambered airfoil
C
              IF(IDEFN.EQ.0)GO TO 110
C
C             IDEFN=1
C             profile defined in terms of Yu(i)=Fu(Xu(i)) , Yl(i)=Fl(Xl(i))
C
C             now to compute YC(I),YPC(I) at XC(I)
C             I=1,NLS+2 is upper surface
C             I=NLS+2,NC is lower surface
C
              M=1;      N=NCORDU;     IND=0;     XCI=XC(NLS+1)
C             T1=FPLEU1(RADLE,XLTU1,YLTU1,XCI)
              T1=(YINU(2)-YINU(1))/(XINU(2)-XINU(1))
              WRITE(5,*)T1
              T2=(YINU(N)-YINU(N-1))/(XINU(N)-XINU(N-1))
              CALL SPLIF(M,N,XINU,YINU,YTNU1,YTNU2,YTNU3,1,T1,1,T2,0,0.0,IND)
              MI=1;     NT=NLS+2
              CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINU,YTNU,YINU1,YINU2,YINU3,0)
              M=1;      N=NCORDL;     IND=0;     XCI=XC(NLS+3)
C             T1=FPLEL1(RADLE,XLTL1,YLTL1,XCI)
              T1=(YINL(2)-YINL(1))/(XINL(2)-XINL(1))
              WRITE(5,*)T1
              T2=(YINL(N)-YINL(N-1))/(XINU(N)-XINU(N-1))
              CALL SPLIF(M,N,XINL,YINL,YINL1,YINL2,YTNL3,1,T1,1,T2,0,0.0,IND)
              MI=NLS+3;         NI=NC
              CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINL,YTNL,YINL1,YINL2,YINL3,0)
C
C             check if the abscissae for the lower and the upper
C             surfaces are the same. if not interpolate to obtain lower
C             surface ordinates at upper aurface abscissae.
C
              IF(NCORDU.NE.NCORDL)GO TO 111
              DO   19   I=1,NCORDU
              IF(ABS(XINL(I)-XINU(I)).GT.1.0E-6)GO TO 111
19            CONTINUE
              DO   18   I=1,NCORDU
              DUMMY(I)=YINL(I)
18            CONTINUE
              GO TO 113
C
C             interpolate to obtain lower surface coordinates at upper surface
C             abscissae
C
111           CONTINUE
              M=1   ;    N=NCORDL  ;    IND=0
              T1=(YINL(2)-YINL(1))/(XINL(2)-XINL(1))
              T2=(YINL(N)-YINL(N-1))/(XINL(N)-XINL(N-1))
              CALL SPLIF(M,N,XINL,YINL,YINL1,YINL2,YTNL3,1,T1,1,T2,0,0.0,IND)
              MI=1   ;    NI=NCORDU
              CALL INTPOL(MI,NI,XINU,DUMMY,DUMMY1,M,N,XINL,YINL,YINL1,YINL2,
             1YINL3,0)
C
C             now compute mean line ordinates by averaging
C
113           CONTINUE
              DO   10   I=1,NCORDU
              YINMN(I)=0.5*(YINU(I)+DUMMY(I))
10            CONTINUE
              M=1;      N=NCORDU;      IND=0
```

```
        T1=(YINMN(2)-YINMN(1))/(XINU(2)-XINU(1))
        T2=(YINMN(N)-YINMN(N-1))/(XINU(N)-XINU(N-1))
        CALL SPLTF(4,N,XTNU,YINMN,YINMN1,YINMN2,YINMN3,1,T1,1,T2,0
       1,0,0,TND)
        MI=1;      NT=NLS+1
        CALL INTPOL(MI,NT,YS,YS,YPS,M,N,XINU,YINMN,YINMN1,YINMN2,YTNMN3
       1,0)
        DO  11  I=1,NLS
        XSI1=XS(I)-XS(I+1)
        YSI1=YS(I)-YS(I+1)
        DS(I)=SQRT(XSI1**2+YSI1**2)
        SINT(I)=YSI1/DS(I);    COST(I)=XSI1/DS(I)
11      CONTINUE
C       replace YC(I),YPC(I) by values computed from approximate
C       analytic expressions for XC(I) < 0.01
C
        CALL FRLE(RADLE,XLEU,YLEU,XLEL,YLEL,ICAM,NLS,NC)
        YPC(NLS+2)=0.0
        GO TO 999
C       -------------------------------------------------------------
110     CONTINUE
C
C       control transfer for IDEFN=0
C       profile defined as Yt(i)=Ft(Xt(i)) , Yc(i)=Fc(Xc(i))
C       YINU(I) stores Yt(i)  and YINL(I) stores Yc(i)
C       XINU(I) stores Xt(i)  and XINL(I) stores Xc(i)
C
        M=1;       N=NCORDL;      IND=0
        T1=(YINL(2)-YINL(1))/(XINL(2)-XINL(1))
        T2=(YINL(N)-YINL(N-1))/(XINL(N)-XINL(N-1))
        CALL SPLTF(M,N,XTNL,YINL,YTNL1,YTNL2,YTNL3,1,T1,1,T2,0,0.0,IND)
        MI=1;      NT=NLS+1
        CALL INTPOL(MI,NI,XS,YS,YPS,M,N,XINL,YTNL,YINL1,TINL2,YINL3,0)
        DO  22  I=1,NLS
        XSI1=XS(I)-XS(I+1)
        YSI1=YS(I)-YS(I+1)
        DS(I)=SQRT(XSI1**2+YSI1**2)
        SINT(I)=YSI1/DS(I)
        COST(I)=XSI1/DS(I)
22      CONTINUE
C
C       now check if the abscissae are the same  for the
C       camber and the thickness distributions
C
        IF(NCORDU.NE.NCORDL)GO TO 114
        DO  23  I=1,NCORDU
        IF(ABS(XINL(I)-XINU(I)).GT.1.0E-6)GO TO 114
23      CONTINUE
        GO TO 115
C
C       abscissae are not the same.  interpolate to obtain camber
C       ordinates at thickness abscissae.
C
114     CONTINUE
        M=1;       N=NCORDL;      IND=0
        MI=1;      NI=NCORDU
        CALL INTPOL(MI,NI,XINU,YINL,DUMMY1,M,N,XINL,YINL,YINL1,YINL2,
       1YINL3,0)
115     CONTINUE
C
C       camber line is now known at the abscissae prescribed for the
C       thickness.  now compute upper and lower surface coordinates
C
        DO  24  I=1,NCORDU
        THETA=ATAN(DUMMY1(I))
        TEMP1=YINL(I)+COS(THETA)*YINU(I)
        TEMP2=XINU(I)-SIN(THETA)*YINU(I)
        TEMP3=XINU(I)+SIN(THETA)*YINU(I)
        YINL(I)=YINL(I)-COS(THETA)*YINU(I)
        XINL(I)=TEMP3
        YINU(I)=TEMP1
        XINU(I)=TEMP2
24      CONTINUE
        M=1;       N=NCORDU;      IND=0
        T1=(YINU(2)-YINU(1))/(XINU(2)-XINU(1))
```

```
00020
00025
00030
00035
00040
00045
00050
00055
00100       T2=(YINU(N)-YINU(N-1))/(XINU(N)-XINU(N-1))
00200       CALL SPLTF(M,N,XINU,YINU,YTNU1,YTNU2,YTNU3,1,T1,1,T2,0,0.0,IND)
00300       MI=1;      NT=NLS+2
00400       CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINU,YTNU,YINU1,YINU2,YINU3,0)
00500       M=1;       N=NCORDU;      IND=0
00600       T1=(YINL(2)-YINL(1))/(XINL(2)-XINL(1))
00700       T2=(YINL(N)-YINL(N-1))/(XINL(N)-XINL(N-1))
00800       CALL SPLTF(M,N,XINL,YINL,YTNL1,YTNL2,YTNL3,1,T1,1,T2,0,0.0,IND)
00900       MI=NLS+3;      NT=NC
01000       CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINL,YTNL,YINL1,YINL2,YINL3,0)
01100 C
01200 C     recompute YC(I),YPC(I) for XC(I) < 0.01
01300 C
01400       CALL FPUC(RADLE,XLEU,YLEU,XLEL,YLEL,TCAM,NLS,NC)
01500       GO TO 999
01600 C     -----------------------------------------------------------------
01700 101   CONTINUE
01800       CLOSE(UNIT=1,DEVICE='DSKC',FILE='AFL.DAT')
01900       CHORD=1.0
02000 C
02100 C     profile prescribed by analytic expressions
02200 C
02300       IF(ICAM.NE.0)GO TO 116
02400 C
02500 C     calculations for a symmetric profile
02600 C
02700       DO   25   I=1,NLS
02800       YS(I)=0.0
02900       DS(I)=YS(I)-YS(I+1)
03000       SINT(I)=0.0;   COST(I)=1.0
03100 25    CONTINUE
03200       DO   26   I=1,NLS+2
03300       IF(I.EQ.NLS+2)GO TO 119
03400       XCI=XC(I)
03500       YC(I)=FPU(XCI)
03600       YPC(I)=FPU1(XCI)
03700       YC(NC-I+1)=-YC(I)
03800       YPC(NC-I+1)=-YPC(I)
03900       GO TO 26
04000 119   YC(I)=0.0;   YPC(I)=0.0
04100 26    CONTINUE
04200       GO  TO  999
04300 C
04400 C     -----------------------------------------------------------------
04500 C
04600 C     calculations for a cambered profile
04700 116   CONTINUE
04800       IF(IDEFN.EQ.0)GO TO 117
04900 C
05000 C     IDEFN=1   ;   profile prescribed as Yu(X)=Fu(X) , Yl(X)=Fl(X)
05100 C
05200       DO   27   I=1,NLS+1
05300       XSI=XS(I)
05400       YU=FPU(XSI);   YL=FPL(XSI)
05500       YS(I)=0.5*(YU+YL)
05600 27    CONTINUE
05700       DO   28   I=1,NLS
05800       XSI1=XS(I)-XS(I+1)
05900       YSI1=YS(I)-YS(I+1)
06000       DS(I)=SQRT(XSI1**2+YSI1**2)
06100       SINT(I)=YSI1/DS(I)
06200       COST(I)=XSI1/DS(I)
06300 28    CONTINUE
06400       DO   29   I=1,NLS+2
06500       IF(I.EQ.NLS+2)GO TO 118
06600       XCI=XC(I)
06700       YC(I)=FPU(XCI)
06800       YPC(I)=FPU1(XCI)
06900       YC(NC-I+1)=FPL(XCI)
07000       YPC(NC-I+1)=FPL1(XCI)
07100       GO TO 29
07200 118   YC(I)=0.0;   YPC(I)=0.0
07300 29    CONTINUE
07400       GO TO 999
07500 C     -----------------------------------------------------------------
```

```
117        CONTINUE
C
C          control transfer for IDEFN=0.
C          profile prescribed analytically in terms of thickness and camber
C          functions  Yt(X)=Ft(X) ,  Yc(X)=Fc(X)
C
           DO  30  I=1,NLS+1
           XSI=XS(I);   YS(I)=FPC(XSI)
30         CONTINUE
           DO  31  I=1,NLS
           XSI1=XS(I)-XS(I+1)
           YSI1=YS(I)-YS(I+1)
           DS(I)=SQRT(XSI1**2+YSI1**2)
           SINT(I)=YSI1/DS(I)
           COSI(I)=XSI1/DS(I)
31         CONTINUE
C
C          DUMMYI is the input abscissae at which upper and lower surface
C          coordinates are computed from the thickness and camber functions
C          X=DUMMYI(I)
C
           DO  32  I=1,NCORDU
           XX=DUMMYI(I);   FPTXX=FPT(XX);      FPCXX=FPC(XX)
           THETA=ATAN(FPC1(XX))
           XINU(I)=XX-SIN(THETA)*FPTXX
           YINU(I)=FPCXX+COS(THETA)*FPTXX
           YINL(I)=FPCXX-COS(THETA)*FPTXX
           XINL(I)=XX+SIN(THETA)*FPTXX
32         CONTINUE
C
C          XINU,YINU,XINL,YINL are now known and interpolation is carried
C          out as before.
C
           M=1;      N=NCORDU;   IND=0
           T1=(YINU(2)-YINU(1))/(XINU(2)-XINU(1))
           T2=(YINU(N)-YINU(N-1))/(XINU(N)-XINU(N-1))
           CALL SPLTF(M,N,XINU,YINU,YTNU1,YTNU2,YTNU3,1,T1,1,T2,0,0.0,IND)
           MI=1;     NT=NLS+2
           CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINU,YTNU,YINU1,YINU2,YINU3,0)
           M=1;      N=NCORDU;   IND=0
           T1=(YINL(2)-YINL(1))/(XINL(2)-XINL(1))
           T2=(YINL(N)-YINL(N-1))/(XINL(N)-XINL(N-1))
           CALL SPLTF(M,N,XINL,YINL,YINL1,YINL2,YINL3,1,T1,1,T2,0,0.0,IND)
           MI=NLS+3;     NT=NC
           CALL INTPOL(MI,NT,XC,YC,YPC,M,N,XINL,YINL,YINL1,YINL2,YINL3,0)
C
C          recompute YC(I),YPC(I) for XC(I) < 0.01
C
999        CONTINUE
C    ------------------------------------------------------------------------
           WRITE(30,501)
           IF(ANALTC)WRITE(30,516)
           IF(.NOT.ANALTC)WRITE(30,517)
           IF(ICAM.EQ.0)WRITE(30,518)
           IF(ICAM.NE.0)WRITE(30,519)
           WRITE(30,520)RADLE,TAU
           IF(IDEFN.EQ.0)WRITE(30,522)
           IF(IDEFN.NE.0)WRITE(30,521)
           IF(.NOT.ANALTC)WRITE(30,523)NCORDU,NCORDL
           WRITE(30,501)
           WRITE(30,502)
           IF(ANALTC)GO TO 998
           WRITE(30,504)(XINU(I),I=1,NCORDU)
           WRITE(30,505)(YINU(I),I=1,NCORDU)
           WRITE(30,514)(XINL(I),I=1,NCORDL)
           WRITE(30,503)(YINL(I),I=1,NCORDL)
998        CONTINUE
           WRITE(30,507)(XS(I),I=1,NLS+1)
           WRITE(30,508)(YS(I),I=1,NLS+1)
           WRITE(30,506)(YPS(I),I=1,NLS+1)
           WRITE(30,509)(DS(I),I=1,NLS)
           WRITE(30,510)(SINT(I),I=1,NLS)
           WRITE(30,511)(COST(I),I=1,NLS)
           WRITE(30,515)(XC(I),I=1,NC)
           WRITE(30,512)(YC(I),I=1,NC)
```

```fortran
      WRITE(30,513)(YPC(I),I=1,NC)
  501 FORMAT(2X,120(1H=))
  502 FORMAT(30X,'results from cubic spline interpolation')
  504 FORMAT(2X,'XTNU(I) values'/(2X,10F10.5))
  505 FORMAT(2X,'YTNU(I) values'/(2X,10F10.5))
  503 FORMAT(2X,'YTNL(I) values'/(2X,10F10.5))
  507 FORMAT(2X,'XS(I) values'/(2X,10F10.5))
  508 FORMAT(2X,'YS(I) values'/(2X,10F10.5))
  506 FORMAT(2X,'YPS(I) values'/(2X,10F10.5))
  509 FORMAT(2X,'DS(I) values'/(2X,10F10.5))
  510 FORMAT(2X,'SINT(I) values'/(2X,10F10.5))
  511 FORMAT(2X,'COST(I) values'/(2X,10F10.5))
  513 FORMAT(2X,'YPC(I) values'/(2X,10F10.5))
  512 FORMAT(2X,'YC(I) values'/(2X,10F10.5))
  514 FORMAT(2X,'XTNL(I) values'/(2X,10F10.5))
  515 FORMAT(2X,'XC(I) values'/(2X,10F10.5))
  516 FORMAT(40X,'profile is prescribed by analytic expressions')
  517 FORMAT(40X,'profile is prescribed at a discrete set of points')
  518 FORMAT(50X,'the airfoil is symmetric')
  519 FORMAT(50X,'the airfoil is cambered')
  520 FORMAT(30X,'the leading edge radius RADLE=',F10.6,5X,'thickness
     1 TAU=',F6.3)
  521 FORMAT(20X,'the airfoil is prescribed in terms of upper and
     1 lower surface distributions')
  522 FORMAT(20X,'the airfoil is prescribed in terms of camber and
     1thickness distributions')
  523 FORMAT(/20X,'for airfoil prescribed in terms of upper and lower
     1 surface coordinates:'/10X,'number of upper surface points NC
     1ORDU=',T3,5X,'number of lower surface coordinates NCORDL=',I3)
  524 FORMAT(2X,'upper and lower surface abscissae do not agree to
     1 within 1.0E-6 at the leading edge')
  525 FORMAT(2X,'upper and lower surface ordinates are not zero at
     1leading  edge')
  526 FORMAT(2X,'upper and lower surface abscissae do not agree to
     1 within 1.0E-6 at the trailing edge')
  527 FORMAT(5X,'the input coordinates as read from file AFL.DAT')
      RETURN
      END
C     **********************************************************
      FUNCTION FPU(X)
      COMMON /E3/ RADLE,IAFCF1,IAFCF2,TAU,BT(10),CT(10)
      IF(X.LE.0.0.OR.X.GT.1.0)FPU=0.0
      IF(X.LE.0.0.OR.X.GT.1.0)RETURN;  X=ABS(X)
      FPU=BT(1)*SQRT(X)+X*(BT(2)+X*(BT(3)+X*(BT(4)+X*BT(5))))
      FPU=TAU*FPU
      RETURN
      END
C     **********************************************************
      FUNCTION FPU1(X)
      COMMON /E3/ RADLE,IAFCF1,IAFCF2,TAU,BT(10),CT(10)
      IF(X.LE.0.0.OR.X.GT.1.0)FPU1=0.0
      IF(X.LE.0.0.OR.X.GT.1.0)RETURN;  X=ABS(X)
      FPU1=0.5*BT(1)/SQRT(X)+BT(2)+X*(2.*BT(3)+X*(3.*BT(4)+X*4.*BT(5))
     1)
      FPU1=TAU*FPU1
      RETURN
      END
C     **********************************************************
      FUNCTION FPL(X)
      FPL=0.0;  RETURN
      END
C     **********************************************************
      FUNCTION FPL1(X)
      FPL1=0.0;  RETURN
      END
C     **********************************************************
      FUNCTION FPC(X)
      FPC=0.0
      RETURN
      END
C     **********************************************************
      FUNCTION FPT(X)
      FPT=0.0
      RETURN
      END
```

```
C       ***************************************************************
        FUNCTION FPC1(X)
        FPC1=0.0
        RETURN
        END
C       ***************************************************************
        SUBROUTINE SPLTF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FQM,IND)
C
C       cubic spline
C       spline is fitted to data array F at nodes S from index M to
C       index N
C       KM=1,2,3-indicates that first second or third derivative is
C       given value VM at point M
C       KN=1,2,3-indicates that first second or third derivative is
C       given value VN at point N
C       IF MODE=0 nodal values of first second and third derivatives
C       of spline are stored in FP,FPP,FPPP arrays so that fitted
C       value at a distance H beyond a node is
C       F+H*FP+H**2*FPP/2.+H**3*FPPP/6.
C       IF MODE > 0 ,FPPP is given the nodal values of the integral of
C       F instead of its third derivative, starting with the value FQM
C       then the third derivative can be recovered as
C       (FPP(I+1)-FP(I))/(S(I+1)-S(I))
C       IND is set equal to zero if S is not a monotone array
C
C       -----------------------------------------------------------------
        DIMENSION S(150),F(150),FP(150),FPP(150),FPPP(150)
        IND=0;    K=IABS(N-M);  IF(K-1)81,81,1
1       K=(N-M)/K;    I=M;      J=M+K
        DS=S(J)-S(I);         D=DS
        IF(DS)11,81,11
11      DF=(F(J)-F(I))/DS
        IF(KM-2)12,13,14
12      U=0.5;    V=3.*(DF-VM)/DS;    GO TO 25
13      U=0.0;    V=VM;               GO TO 25
14      U=-1.0;   V=-DS*VM;           GO TO 25
21      I=J;      J=J+K;    DS=S(J)-S(I);    IF(D*DS)81,81,23
23      DF=(F(J)-F(I))/DS; B=1./(DS+DS+U); U=B*DS; V=B*(6.*DF-V)
25      FP(I)=U;   FPP(I)=V;   U=(2.-U)*DS
        V=6.*DF+DS*V;       IF(J-N)21,31,21
31      IF(KN-2)32,33,34
32      V=(6.*VN-V)/U;            GO TO 35
33      V=VN;                     GO TO 35
34      V=(DS*VN+FPP(I))/(1.+FP(I))
35      B=V;               D=DS
41      DS=S(J)-S(I);      U=FPP(I)-FP(I)*V;   FPPP(I)=(V-U)/DS
        FPP(I)=U;       FP(I)=(F(J)-F(I))/DS-DS*(V+U+U)/6.
        V=U;      J=I;       I=I-K;      IF(J-M)41,51,41
51      I=N-K;       FPPP(N)=FPPP(I);   FPP(N)=B;      IND=1
        FP(N)=DF+D*(FPP(I)+B+B)/6.;              IF(MODE)81,81,61
61      FPPP(J)=FQM;                    V=FPP(J)
71      I=J;   J=J+K;   DS=S(J)-S(I);   U=FPP(J)
        FPPP(J)=FPPP(I)+0.5*DS*(F(I)+F(J)-DS*DS*(U+V)/12.0)
        V=U;            IF(J-N)71,81,71
81      RETURN
        END
C       ***************************************************************
        SUBROUTINE INTPOL(MI,NI,SI,FI,FIP,M,N,S,F,FP,FPP,FPPP,MODE)
C
C       interpolation using piecewise TAYLOR's series as generated by
C       cubic spline or its integral.  values F,FP,FPP,FPPP of function
C       and its derivatives are given at its nodes S from index MI to NI
C       if MODE > 0 a correction is added for a piecewise constant
C       fourth derivative so that integral of cubic spline is evaluated
C       exactly.
C
        DIMENSION SI(150),FI(150),FIP(150),S(150),F(150),FP(150),
       1FPP(150),FPPP(150)
        K=IABS(N-M);    K=(N-M)/K;    I=M;    MIN=MI
        NIN=NI;   D=S(N)-S(M);   IF(D*(SI(NI)-SI(MI)))11,13,13
11      MIN=NI;   NIN=MI
13      KI=IABS(NIN-MIN);    IF(KI)21,21,15
15      KI=(NIN-MIN)/KI
21      II=MIN-KI;         C=0.0;    IF(MODE)31,31,23
23      C=1.0
```

```
31      TI=IT+KU;           SS=S(IT)
33      T=T+K;              IF(I-1)35,37,35
35      IF(D*(S(T)-SS))33,33,37
37      T=T;    T=T-K;      SS=SS-S(I)
        FPPPP=C*(FPPP(I)-FPPP(T))/(S(I)-S(T))
        FF=FPPP(T)+0.25*SS*FPPPP;        FF=FPP(I)+SS*FF/3.
        FF=FP(T)+0.5*SS*FF;    FI(TI)=F(T)+SS*FF
        FIP(TI)=FP(I)+SS*(FPP(T)+0.5*SS*(FPPP(T)+SS*FPPPP/3.))
        IF(IT-MEM)31,41,31
41      RETURN
        END
C       *************************************************************
        SUBROUTINE FPLE(RADLE,XLEU,YLEU,XLEL,YLEL,ICAM,NLS,NC)
        DIMENSION XLEU(3),YLEU(3),XLEL(3),YLEL(3)
        DIMENSION AU(3),AL(3)
        COMMON /C2/ XC(55),YC(55),YPC(55)
        YUP(X) =AU(1)*SQRT(X)+AU(2)*X+AU(3)*X**2
        YUP1(X)=0.5*AU(1)/SQRT(X)+AU(2)+2.*AU(3)*X
        YLO(X) =AL(1)*SQRT(X)+AL(2)*X+AL(3)*X**2
        YLO1(X)=0.5*AL(1)/SQRT(X)+AL(2)+2.*AL(3)*X
        AU(1)=SQRT(2.*RADLE);    AL(1)=-AU(1)
        XMAXU=XLEU(2);   XMAXL=XLEL(2)
        X1=XLEU(1)           ;           X2=XLEU(2)
        X12=X1*X2            ;           X22=X2**2
        B1=YLEU(1)-AU(1)*SQRT(X1)
        B2=YLEU(2)-AU(1)*SQRT(X2)
        B=B1*X22-B2*X12
        C=B1*X2-B2*X1
        D=X12*X2-X22*X1
        AU(2)=-B/D;      AU(3)=C/D
        X1=XLEL(1)           ;           X2=XLEL(2)
        X12=X1*X2            ;           X22=X2**2
        B1=YLEL(1)-AL(1)*SQRT(X1)
        B2=YLEL(2)-AL(1)*SQRT(X2)
        B=B1*X22-B2*X12
        C=B1*X2-B2*X1
        D=X12*X2-X22*X1
        AL(2)=-B/D;      AL(3)=C/D
        DO 10   I=1,NLS+1
        XCT=XC(I);       IF(XCI.GT.0.01)GO TO 102
        YC(I)=YUP(XCT)
        YPC(I)=YUP1(XCT)
        IF(ICAM.EQ.0)GO TO 101
102     CONTINUE
        IF(XCI.GT.0.01)GO TO 10
        YC(NC-I+1)=YLO(XCI)
        YPC(NC-I+1)=YLO1(XCI)
        GO TO 10
101     YC(NC-I+1)=-YC(I)
        YPC(NC-I+1)=-YPC(I)
10      CONTINUE
        YC(NLS+2)=0.0;   YPC(NLS+2)=0.0
        RETURN
        END
C       *************************************************************
C       *************************************************************
        FUNCTION FPLEU1(RADLE,XLTU1,YLTU1,X)
        AOU=SQRT(2.*RADLE)
        A1U=(YLTU1-AOU*SQRT(XLTU1))/XLTU1
        FPLEU1=0.5*AOU/SQRT(X)+A1U
        RETURN
        END
C       *************************************************************
        FUNCTION FPLEL1(RADLE,XLTL1,YLTL1,X)
        AOL=-SQRT(2.*RADLE)
        A1L=(YLTL1-AOL*SQRT(XLTL1))/XLTL1
        FPLEL1=0.5*AOL/SQRT(X)+A1L
        RETURN
        END
C       *************************************************************
C       *************************************************************
```

```
C     THIS IS FILE AMFAVF.FOR
C
C     SUBROUTINE AMFAVF(NLS,NC,NXI,NI,NMAX,MMAX,NJLAST,ICAM)
C
C     This subroutine calculates the influence coefficients due to
C     the internal singularities.
      COMPLEX ZGRID,ZNET
      COMMON /C3/ AS(23,51,5),BS(23,51,5),CV(23,51,5),DV(23,51,5)
      COMMON /C1/ XS(55),YS(55),YPS(55),DS(55),SINT(55),COST(55)
      COMMON /C2/ XC(55),YC(55),YPC(55),DSP(55)
      COMMON /C6/ ZGRID(52,6),ZNET(52,6)
      PIR=3.14159265358979323; PIF=0.5/PIR
      DO 10  J=1,MMAX
      DO  11  I=1,NMAX
      XGIJ=REAL(ZGRID(I,J))
      YGIJ=AIMAG(ZGRID(I,J))
      AIKL=0.0;  CIKL=0.0
      DO  12  K=1,NLS+2
      IF(K.GT.NLS)GO TO 101
      XSKM=0.5*(XS(K+1)+XS(K))
      YSKM=0.5*(YS(K+1)+YS(K))
      XTK=((XGIJ-XSKM)*COST(K)+(YGIJ-YSKM)*SINT(K))
      YTK=(-(XGIJ-XSKM)*SINT(K)+(YGIJ-YSKM)*COST(K))
      FIK=FC(XTK,YTK,DS(K));  GIK=GC(XTK,YTK,DS(K))
      FIK=PIF*FIK;  GIK=PIF*GIK
      AS(K,I,J)=FIK*COST(K)-GIK*SINT(K)
      BS(K,I,J)=FIK*SINT(K)+GIK*COST(K)
      GTKH=0.5*GTK;  FTKH=0.5*FIK
      FIKXTK=FTK*XTK/DS(K);  FIKYIK=FIK*YIK/DS(K)
      GIKXTK=GTK*XTK/DS(K);  GIKYIK=GIK*YIK/DS(K)
      AIK1P=(GTKH+FIKYTK-GTKXIK)*COST(K)
      AIK2P=(FTKH-FIKXTK-GTKYIK+PIF)*SINT(K)
      AIKP=ATK1P+ATK2P
      BIK1P=(GTKH-FIKYTK+GTKXIK)*COST(K)
      BIK2P=(FTKH+FIKXTK+GTKYIK-PIF)*SINT(K)
      BIKP=BTK1P+BTK2P
      CIK1P=(GTKH+FIKYTK-GTKXIK)*SINT(K)
      CIK2P=(FTKH-FIKXTK-GTKYIK+PIF)*COST(K)
      CIKP=CIK1P-CIK2P
      DIK1P=(GTKH-FIKYIK+GIKYIK-PIF)*SINT(K)
      DIK2P=(FTKH+FIKXTK+GIKYIK-PIF)*COST(K)
      DIKP=DTK1P-DTK2P
      CV(K,I,J)=AIKL+BIKP;  DV(K,I,J)=CIKL+DTKP
      AIKL=AIKP;  CIKL=CTKP
      IF(K.EQ.NLS)CV(K+1,I,J)=AIKP
      IF(K.EQ.NLS)DV(K+1,I,J)=CIKP
      GO TO 12
101   XCXS=XGIJ-XS(K-1)
      YCYS=YGIJ-YS(K-1)
      RR=XCXS**2+YCYS**2
      AS(K,I,J)=PIF*XCXS/RR;  BS(K,I,J)=PIF*YCYS/RR
      IF(K.EQ.NLS+1)GO TO 12
      XCXS=XGIJ-0.5*(XS(NLS)+XS(NLS+1))
      YCYS=YGIJ-0.5*(YS(NLS)+YS(NLS+1))
```

```
                RR=XCXS**2+YCYS**2
                CV(K,I,J)=PIF*YCYS/RR;   DV(K,I,J)=-PIF*XCXS/RR
12              CONTINUE
11              CONTINUE
10              RETURN
                OPEN(UNIT=24,DEVICE='DSKD')
                DO  13   K=1,NGS+2
                WRITE(24,506)((AS(K,I,J),I=1,NMAX),J=1,MMAX)
                WRITE(24,506)((BS(K,I,J),I=1,NMAX),J=1,MMAX)
                WRITE(24,506)((CV(K,I,J),I=1,NMAX),J=1,MMAX)
                WRITE(24,506)((DV(K,I,J),I=1,NMAX),J=1,MMAX)
13              CONTINUE
506             FORMAT(10E13.7)
                RETURN
                END
C       ****************************************************************
                FUNCTION FC(X,Z,DEL)
                DEL=0.5*DEL;   Z2=Z**2
                FC=0.5*ALOG(((X+DEL)**2+Z2)/((X-DEL)**2+Z2))
                RETURN
                END
C       ****************************************************************
                FUNCTION GC(X,Z,DEL)
                DEL=0.5*DEL
                IF(ABS(Z).LE.0.000001)GO TO 101
                GC=(ATAN((X+DEL)/Z)-ATAN((X-DEL)/Z));   RETURN
101             GC=0.0;   RETURN
                END
C       ****************************************************************
```

```
C
C      THIS IS FILE ARTVS1.FOR
C
C      SUBROUTINE ARTVIS
C
C      this subroutine adds the artificial viscosity term to GF(I,J)
C      and returns the modified value to the MAIN routine.
C
       COMMON /C1/ NUS,NMAX,MMAX,NW,NC,ICAM,IALFA
       COMMON /C2/ FSM,ALFA,GAM,EXPGAM,FGAM,PRF,QSONIC,SK,BETA2,LAMAX
       COMMON /C5/ CSALFA,SNALFA
       COMMON /C8/ RHO(51,5)
       COMMON /C10/ UF(51,5),VF(51,5)
       COMMON /C12/ Q(51,5),QS(51,5),RMU(51,5)
       COMMON /C14/ GFN(51,5),R(51,5),GF(51,5)
       COMMON /C19/ A2(51,5),RMACH(51,5)
       COMMON /C20/ Q2(51,5),Q2X(51,5),Q2Y(51,5)
       REAL LAMAX
C      ---------------------------------------------------------------
       IMID=(MMAX+1)/2
       DO  10   I=1,MMAX
       DO  11   I=1,NMAX
       IF(I.LT.NW.OR.I.GT.NW+NC)GO TO 11
       IF(Q(I,J).LT.QSONIC.AND.Q(I+1,J).LT.QSONIC.AND.Q(I-1,J).LT.
      1QSONIC)GO TO 11
       UFIP1J=CSALFA+UF(I+1,J);   UFIJ=CSALFA+UF(I,J)
       UFIM1J=CSALFA+UF(I-1,J)
       RBA2=RHO(I,J)/A2(I,J);      RBA2P1=RHO(I+1,J)/A2(I+1,J)
       RBA2M1=RHO(I-1,J)/A2(I-1,J)
       IF(I.GT.IMID)GO TO 101
       TIJ1=(UFIP1J*RBA2P1*RMU(I+1,J)*Q2X(I+1,J)-UFIJ*RMU(I,J)*
      1RBA2*Q2X(I,J))
       TIJ=LAMAX*TIJ1
       GFN(I,J)=GFN(I,J)+TIJ
       GO TO 11
  101  TIJ1=-(UFIJ*RBA2*RMU(I,J)*Q2X(I,J)-UFIM1J*RMU(I-1,J)*
      1RBA2M1*Q2X(I-1,J))
       TIJ=LAMAX*TIJ1
       GFN(I,J)=GFN(I,J)+TIJ
C      ---------------------------------------------------------------
  11   CONTINUE
  10   CONTINUE
       RETURN
       END
C
       ***********************************************************************
```

```
C       THIS IS FILE GRID.FOR
C
        SUBROUTINE GRID(NBLS,NTLS,NALS,NW,NJ,JC,SPI,NJLAST,BETMIN,ORIGIN
       1,DELX,EPSTE,XSHL,XSHR,ICAM,NMAX,IMAX,NLS,NC,DELTA,CHORD)
C
C       This program together with subroutines AERFOL and HESS
C       contained in files AERFOL.FOR and HESS.FOR respectively,
C       calculates the field influence coefficients.
C       GRID.DAT contains the input data for this program
C       the command is:EX GRID,AERFOL,HESS;
C
        PARAMETER CHORD=*(2,0)
        DIMENSION DELTAXI(55),GTAYIG(55)
        DIMENSION Z(55),W(55)
        DIMENSION XGRID(52,6),YGRID(52,6)
        DIMENSION XIS(55),YIS(55),BETAS(10),BETAN(10),BTAXIN(55)
        COMMON /B1/ BAULD,TAPER1,TAPER2,TAU,BT(10),CT(10)
        COMMON /B2/ XS(55),YS(55),YPS(55),DS(55),SINP(55),COSP(55)
        COMMON /B3/ XC(55),YC(55),YPC(55),DSP(55)
        COMMON /B4/ ZGRID(52,6),ZIGT(52,6)
        COMMON /B13/ DX1(55),DST(10)
        COMMON /B21/ DELXIJ(51,5,2,2)
        COMMON /B21/ DELXTJ(51,5)
        ZETA(Z)=1.-4.*CEXP(Z)
        OPEN(UNIT=1,DEVICE='DSKC',FILE='GRID.DAT')
        READ (1,*) NBLS,NTLS,NALS,NW,NJ,JC,SPI,NJLAST,BETMIN,DELX,
       1XSHL,YSHR,DELTA,EPSTE
        PIR=3.141592653589793238; ZPI=CMPLX(0.,2.*PIR)
        NLS=NBLS+NTLS+NALS
        NJ1=NJ+1;   NX1=2*(NLS+NW)+3
        NMAX=NXG;   IMAX=NJ-NJLAST;   NC=2*NLS+3
        XS(1)=1.0;   XS(2)=1.0-EPSTE
        DXLS=(1.0-EPSTE-XSHR)/FLOAT(NALS-1)
        DO 11   I=3,NALS+1
        XS(I)=XS(I-1)-DXLS
11      CONTINUE
        NST=NALS+NTLS+1
        DXLS=(XSHR-XSHL)/FLOAT(NTLS)
        DO 12   I=NALS+2,NST
        XS(I)=XS(I-1)-DXLS
12      CONTINUE
        RINT=XSHL-DELTA
        DO 37   I=NST+1,NLS+1
        THETA=0.5*PIR*FLOAT(I-NST)/FLOAT(NLS-NST+1)
        XX=RINT*SIN(THETA)
        XS(I)=XSHL-XX
37      CONTINUE
        DO 13   I=1,NLS
        XC(I)=0.5*(XS(I)+XS(I+1))
13      CONTINUE
        XC(NLS+1)=0.5*XS(NLS+1)
        XC(NLS+2)=0.0
        XC(NLS+3)=XC(NLS+1)
        DO 14   I=NLS+4,NC
        XS(I)=XC(NC-I+1)
14      CONTINUE
        CALL AERFOL(NLS,NC,ICAM,DELTA,ORIGIN,CHORD)
        DO 16   I=1,NLS+1
        Z(I)=CMPLX(XC(I)-ORIGIN,YC(I))
        IF(ICAM.NE.0)Z(NC-I+1)=CMPLX(XC(NC-I+1)-ORIGIN,YC(NC-I+1))
        IF(ICAM.EQ.0)Z(NC-I+1)=CONJG(Z(I))
16      CONTINUE
        Z(NLS+2)=CMPLX(-ORIGIN,0.0)
        DO 17   I=1,NC
        ZETA1=ZETA(Z(I))
        ZETA2=CSQRT(ZETA1**2-1.0)
        W(I)=CLOG(ZETA1+ZETA2)
17      CONTINUE
C       ----------------------------------------------------------------
C       Inverse transformation
C       ----------------------------------------------------------------
        DO 18   I=1,NC
        WCOSH=(CEXP(W(I))+CEXP(-W(I)))/2.
```

```
      ZRDG=(1.+ZCOSH)/2.
      ZINV=CLOG(ZRDG)
      IF(REAL(W(I)).GT.0.0)ZINV=Z(NV-ZPI
      ZDPROR=Z(I)-ZINV
      IF(CABS(ZDPROR).LT.0.0001)GO TO 18
      WRITE(2,708)
708   FORMAT(2X,'error in inverse transformation, program is
     1 terminated, the current values of I,Z(I),ZINV are'/2X,'I=',
     1 I2,2X,2(F10.5,',',F10.5,3X))
18    CONTINUE
C     --------------------------------------------------------------
C     calculation of grid points in (xi,eta) plane and trans-
C     -formation to (x,y) plane.
C     --------------------------------------------------------------
      XITE=1.0;      YTE=0.0002
      ZIP=CMPLX(XITE-DXTGTX,(TG)
      ZETA1=ZETA(ZITF);    ZETA2=CSQRT(ZETA1**2-1.0)
      WIP=CLOG(ZETA1+ZETA2)
      XITEL=REAL(WTG);    XITER=XITEL
      ETATEL=AIMAG(WTG);   ETATER=ETATEL
      DXITE=ABS(REAL(W(1))-XITEL);   DXITEL=10.*DXITE
      IF(NW.EQ.0)GO TO 101
      XIG(NW)=XITEL-DXITE
      ETAXTG(NW)=ETATEL
      DXINC=DXITE
      DO 19    I=1,NW-1
      XIG(NW-I)=XIG(NW-I+1)-DXINC
      ETAXTG(NW-I)=ETATEL
      DXINC=2.*DXINC
19    CONTINUE
101   CONTINUE
      DO 20    I=1,NC
      XIG(NW+I)=REAL(W(I))
      ETAXTG(NW+I)=AIMAG(W(I))
20    CONTINUE
      IF(NW.EQ.0)GO TO 102
      DO 21    I=1,NW
      XIG(NW+NC+I)=-XIG(NW-I+1)
      ETAXTG(NW+NC+I)=ETAXTG(NW-I+1)
21    CONTINUE
102   CONTINUE
      DO 25    I=1,NXT-1
      DXT(I)=XIG(I+1)-XIG(I)
25    CONTINUE
      DXINC=0.5*(XIG(NLS+2+NW)-XIG(NLS+1+NW))
      XIN(NLS+3+NW)=XIG(NLS+2+NW)+DXINC
      ETAXIN(NLS+3+NW)=0.5*(ETAXTG(NLS+3+NW)+ETAXIG(NLS+2+NW))
      DO 34    I=1,NLS
      XIN(NLS+3+NW-I)=0.5*(XIG(NLS+3+NW-I)+XTG(NLS+2+NW-I))
      ETAXIN(NLS+3+NW-I)=0.5*(ETAXTG(NLS+3+NW-I)+ETAXIG(NLS+2+NW-I))
34    CONTINUE
      XIN(NW+1)=XITEL;    ETAXIN(NW+1)=ETATEL
      XIN(NW+2)=XIG(NW+1)+DXITE
      ETAXIN(NW+2)=ETATEL-2.*(ETATEL-ETAXIG(NW+1))
      IF(NW.EQ.0)GO TO 103
      XIN(NW)=XIG(NW)-DXITE;    ETAXIN(NW)=ETATEL
      DXINC=XIN(NW)-XIG(NW-1)
      DO 26    I=1,NW-1
      XIN(NW-I)=XIG(NW-I)-DXINC
      ETAXIN(NW-I)=ETATEL
      IF(L.GT.NW-2)DXINC=XIN(NW-I)-XIG(NW-I-1)
26    CONTINUE
103   CONTINUE
      DO 35    I=NW+NLS+4,NXT-NW-1
      XIN(I)=0.5*(XIG(I-1)+XIG(I))
      ETAXIN(I)=0.5*(ETAXIG(I-1)+ETAXIG(I))
35    CONTINUE
      XIN(NXT-NW)=XIG(NXT-NW)-DXITE
      ETAXIN(NXT-NW)=ETATER-2.*(ETATER-ETAXIG(NXT-NW))
      XIN(NXT-NW+1)=XITER
      ETAXIN(NXT-NW+1)=ETATER
      IF(NW.EQ.0)GO TO 104
      DO 36    I=NXT-NW+2,NXT+1
      XIN(I)=-XIN(NXT+2-I)
      ETAXIN(I)=ETAXIN(NXT+2-I)
36    CONTINUE
```

```
104      CONTINUE
         DO 27   I=1,NXT
27       DETAN(I)=(ETAXIN(I+1)-ETAXIN(I))/(XIN(I+1)-XIN(I))
         DO 22   J=1,JC
         RETAN(J)=(1.-SFI*ID)/(SFI*SI+IC*(1.-SFI)-0.5)
         DO 22   I=K,JC+1
         BETAN(I)=BETAN(J-1)-DELBET
         IF(J.EQ.1)BETAN(I)=BETAN(J-1)-0.5*DELBET
22       CONTINUE
         DELBET=SFI*DELBET
         DO 39   I=JC+2,JJ
39       BETAN(I)=BETAN(J-1)-DELBET
         DO 23   I=1,JJ
         BETAG(I)=0.5*(BETAN(I)+BETAN(I+1))
         IF(J.EQ.1)BETAG(I)=BETAN(J)
23       CONTINUE
         DO 24   I=1,JJ-1
         DBET(I)=BETAG(I+1)-BETAG(J)
24       CONTINUE
         DO 28   I=1,NXT+1
         XINI=XIN(I);   ETYXIGI=ETAXIG(I)
         XINI=XIN(A);   ETYXINI=ETAXIN(I)
         DO 29   J=1,JJ1
         ETANJ=BETAN(I)*ETXTNT
         WTN=CMPLX(XINI,ETANJ);    WNET(I,J)=WTN
         WCOSH=(CEXP(W(N)+CEXP(-WTN))/2.
         ZARG=(1.-WCOSH)/2.
         ZZ=CLOG(ZARG)
         IF(REAL(WIN).GT.0.0)ZZ=ZZ-ZPI
         ZNET(I,J)=ZZ+ORIGIN
         IF(J.EQ.1.AND.I.GE.NW+NC+1)ZNET(I,J)=CONJG(ZNET(2*NW+NC+2-I,J))
         IF(J.EQ.NJ1)GO TO 29
         BETAGI=BETAG(I)*ETXTGT
         WTG=CMPLX(XIGI,ETAGJ)
         WGRID(I,J)=WTG
         WCOSH=(CEXP(WTG)+CEXP(-WTG))/2.
         ZARG=(1.-WCOSH)/2.;   ZZ=CLOG(ZARG)
         IF(REAL(WTG).GT.0.0)ZZ=ZZ-ZPI
         ZGRID(I,J)=ZZ+ORIGIN
         IF(J.EQ.1.AND.I.GT.NW+NC)ZGRID(I,J)=CONJG(ZGRID(2*NW+NC+1-I,J))
29       CONTINUE
28       CONTINUE
         DO 40   I=1,NMAX-1
         X1=REAL(ZGRID(I,1))-REAL(ZGRID(I+1,1))
         Y1=AIMAG(ZGRID(I,1))-AIMAG(ZGRID(I+1,1))
         IF(I.LE.NDS+2+NW)DSP(I)=-SQRT(X1**2+Y1**2)
         IF(I.GT.NDS+2+NW)DSP(I)=SQRT(X1**2+Y1**2)
40       CONTINUE
         IMID=(NMAX+1)/2
         DO 41   J=1,MMAX
         DO 41   I=1,NMAX
         IF(I.LT.IMID)DELXIJ(I,J)=-REAL(ZGRID(I+1,J)-ZGRID(I,J))
         IF(I.GE.IMID)DELXIJ(I,J)=REAL(ZGRID(I+1,J)-ZGRID(I,J))
41       CONTINUE
C
C        calculate the Influence coefficients----------------------------
         CALL HESS(NXT,NJ,NMAX,MMAX,NJLAST,TCAM)
         STOP
C
C        STOP------------------------------------------------------------
         END
C        ****************************************************************
```

```
      SUBROUTINE GRSS(DAT,AJ,NMAX,MMAX,NLAST,ICAR)

C     This subroutine calculates the field influence coefficients
C     PSU and PSV and writes on the disk in files FOR22.DAT and
C     FOR23.DAT.

      COMPLEX ZGRID,ZUFT
      DIMENSION PSV(4),PSV(4)
      DIMENSION X(5),Y(5)
      COMMON /C5/ ZGRID(52,6),ZUFT(52,6)
      COMMON /C6/ PSV(51,26,5),PSU(51,5,26,5)
      PI=3.14159265358979323;  DAIRPI=0.25/PIR
      KK=KMAX;  ICOUNT=1;  IMID=(MX1+1)/2
      IF(ISV.EQ.1)KMAX=(MX1+1)/2
      DO  11  L=1,MMAX
      DO  11  K=1,KMAX
      XC=REAL(ZGRID(K,L));  YC=AIMAG(ZGRID(K,L))
      KK=2
      DO  12  LI=L,L+1
      DO  13  KI=K,K+1
      X(KK)=REAL(ZUFT(KI,LI));  Y(KK)=AIMAG(ZUFT(KI,LI))
      IF(GT.EQ.L)KK=KK-1;  IF(LI.EQ.L+1)KK=KK+1
      CONTINUE;  KK=2
   13 CONTINUE
   12 CONTINUE
      D31=SQRT((X(1)-X(3))**2+(Y(1)-Y(3))**2)
      D42=SQRT((X(4)-X(2))**2+(Y(4)-Y(2))**2)
      DMAX=D31;  IF(D42.GT.D31)DMAX=D42
      D12=SQRT((X(1)-X(2))**2+(Y(1)-Y(2))**2)
      D23=SQRT((X(2)-X(3))**2+(Y(2)-Y(3))**2)
      D14=SQRT((X(1)-X(4))**2+(Y(1)-Y(4))**2)
      D43=SQRT((X(4)-X(3))**2+(Y(4)-Y(3))**2)
      SP1=0.5*(D12+D23+D31);  SP2=0.5*(D14+D43+D31)
      AREA1=SQRT(SP1*(SP1-D12)*(SP1-D23)*(SP1-D31))
      AREA2=SQRT(SP2*(SP2-D14)*(SP2-D43)*(SP2-D31))
      AREA=AREA1+AREA2
      TEMPX=X(2);  TEMPY=Y(2)
      X(2)=X(4);  Y(2)=Y(4)
      X(4)=TEMPX;  Y(4)=TEMPY
      DO  14  I=1,4
      X(I)=X(I)-XC;  Y(I)=Y(I)-YC
   14 CONTINUE
      DO  15  J=1,MMAX
      IMAX=NMAX
      DO  16  I=1,IMAX
      XIJ=REAL(ZGRID(I,J))
      YIJ=AIMAG(ZGRID(I,J))
      XI=XIJ-XC;  YJ=YIJ-YC
      XI2=XI**2;  YJ2=YJ**2
      IF(SQRT(XI2+YJ2).GT.4.*DMAX)GO TO 103
      DO  19  MU=1,4
      NU=MU+1;  IF(MU.EQ.4)NU=1
      XNU=X(NU);  YNU=Y(NU)
      XMU=X(MU);  YMU=Y(MU)
      CONDA=ABS(YNU-YMU);  CONDB=ABS(XNU-XMU)
      IF(CONDA.GT.0.00001.AND.CONDB.GT.0.00001)GO TO 101
      IF(CONDA.GT.0.00001)GO TO 101
      FJU(MU)=0.0;  IF(CONDB.GT.0.00001)GO TO 102; GO TO 104
  101 QMN=(XNU-XMU)/(YNU-YMU)
      PMN=-QMN*YMU+XMU
      E3MN2=1.+QMN**2
      E3MN=SQRT(E3MN2)
      E1MN2=((XI-PMN)**2+YJ2)/E3MN2
      E1MN=SQRT(E1MN2)
      E2MN=-((XI-PMN)*QMN+YJ)/E3MN2
      E2MN2=E2MN**2
      E4MN2=E1MN2-E2MN2
      IF(E4MN2.LT.0.0)WRITE(5,510)I,J,K,L,E4MN2
      E4MN=SQRT(E4MN2)
      FJU(MU)=((YNU-YMU)*ALOG(E3MN2)+FUN1(YNU,YMU,E2MN,E4MN))
      IF(CONDB.GT.0.00001)GO TO 104
      IF(J.GT.4)GO TO 19
  102 CONTINUE
      IF(J.GT.4)GO TO 19
      SMN=(YNU-YMU)/(XNU-XMU)
```

```
              S=SIN(FSG1+YI2)
              G2=1.+SIN(**2
              G3=SQRT(G3MR2)
        ...  I2=(XI2+YJ)-R**1**2)/G3MR2
        ...  =SQRT(G3MR2)
        ...  =*(XI+YJ)-R**1)*SIN1/G3MR2
        ...  2=E(XI)**2)
        ...  2=G3MR2-G4MR2
              IF(G4MR2.GT.0.0)WRITE(5,510)I,J,K,L,G4MR2
              G4M=SQRT(G4MR2)
        FUU(J)=E((XMU-XMU)*ALOG(G3MR2)+FUN1(XMU,XMU,G2MN,G4MN))
              ...
       FVV(J)=...
          CONTINUE
       FSU(I,J,K,L)=DATRPI*(FUU(1)+FUU(2)+FUU(3)+FUU(4))
       FSV(I,J,K,L)=-DATRPI*(FVV(1)+FVV(2)+FVV(3)+FVV(4))
              ...
       FSU(I,J,K,L)=2.*DATRPI*XI*AREA/(XI2+YJ2)
       FSV(I,J,K,L)=2.*DATRPI*YJ*AREA/(XI2+YJ2)
          CONTINUE
          CONTINUE
          CONTINUE
          CONTINUE
       WRITE(5,505)L
       FORMAT(2X,'calculations completed for row L=',I2)
          CONTINUE
       WRITE(5,*)ICOUNT
       OPEN(UNIT=22,DEVICE='DSK')
       OPEN(UNIT=23,DEVICE='DSK')
       DO 17 I=1,IMAX
       DO 18 J=1,JMAX
       WRITE(22,506)((FSU(I,J,K,L),K=1,KMAX),L=1,MMAX)
       IF(J.GT.1)GO TO 18
       WRITE(23,506)((FSV(I,K,L),K=1,KMAX),L=1,MMAX)
18     CONTINUE
17     CONTINUE
506    FORMAT(10F13.8)
C506   FORMAT(2X,'K=',I2,2X,'L=',I2/(2X,10F12.8))
510    FORMAT(2X,'I,J,K,L are',4I3,2X,2E15.7)
       RETURN
       END
C      **********************************************************************
       FUNCTION FUN1(A,B,C,D)
       D2=D**2; APC=A+C; BPC=B+C
       DD=D; IF(D.LT.1.0E-6)DD=1.0E-6
       APC2=APC**2; BPC2=BPC**2
       F1=APC*ALOG(APC2+D2)-BPC*ALOG(BPC2+D2)-2.*(A-B)
       F2=2.*D*(ATAN(APC/DD)-ATAN(BPC/DD))
       FUN1=F1+F2
       RETURN
       END
C      **********************************************************************
```

```
5,1
4,11,3,3,5,3,2.0,0,0.50,0.001,0.10,0.80,0.0078,0.02
0.0156,0.12,0.1
```

THIS IS DATA FILE "GRID.DAT" FOR PROGRAM PACKAGE GRID.FOR
THIS DATA LINE IS THE SAME AS THE THIRD LINE IN FILE "B1.DAT"

```
4,11,3,3,5,3,2.0,0,0.50,0.001,0.10,0.80,0.0078,0.02
```

THIS FILE IS "AFL.DAT" FOR DATA READ IN SUBROUTINE AERFOL
THIS DATA IS FOR A NACA 0012 AIRFOIL
FOR OTHER AIRFOILS SEE SUBROUTINE AERFOL FOR DATA INPUT

```
5,1
0.0156,0.12,0.1
5,1.4845,-0.6300,-1.7580,1.4215,-0.5075
```

```
**************************************************************************
**************************************************************************
**************************************************************************

        TWO-DIMENSIONAL TRANSONIC FLOW PAST PLANE AIRFOILS
    USING FULL-POTENTIAL EQUATION AND EXACT TANGENCY BOUNDARY
        CONDITION BY THE INTEGRAL EQUATION METHOD.


            The following is the input data specification

**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************


    AVRG=   0

    FSM= 0.750    ALFA= 0.000   GAM= 1.400    LAMAX= 4.000

    NBLS=   1      NILS= 14      NALS=  3       NW=  3

    NI=  5      IC=  3      SFJ= 2.00     NILAST=  0

    BETMIN= 0.500      DELX= 0.0010      DELTA= 0.0078000

    XSHL= 0.100     XSHR= 0.800      EPSTE= 0.020

    EPS= 0.035      ITMAX=  3      ALP1= 0.500      ALP2= 0.000


    ****************************************************************************
    ****************************************************************************


        calculations without GF(I,J) averaging
        for the first row of field panels

        airfoil specification:
        viscosity scheme:

        free stream mach number FSM= .750
        airfoil thickness TAU= .120
        angle of attack ALFAD= 2.00
        artificial viscosity factor LAMAX= 4.000

        number of line  elements NLS= 21
        number of collocation points NC= 45
        number of points before expected fore shock location NBLC= 6
        after downstream shock location NALC= 3
        number of points in the expected shock interval NILC=14
        shock interval XSHL< X < XSHR  XSHL= 0.100  XSHR= 0.800

        GRID DETAILS:  NMAX= 51       MMAX=  5          NW=  3
                       XMAX=  1.44    XMIN= -0.78     YMAX=  1.44

        sonic velocity QSONIC=  1.284
        sonic pressure coefficient CPSTAR= -0.591



    **************************************************************************
    **************************************************************************
    **************************************************************************
```

COMPRESSIBLE FLOW RESULTS

lift coeff CL= 0.3052
drag coeff CD= 0.0012
moment coef about C= 0.00 is CM= -0.0630?

```
   X/C(I)           CP(I,M)(I)
   .9900           -0.2358
   .9750           -0.0608
   .9750           -0.0383
   .7750           -0.0987
   .7250           -0.1361
   .6850           -0.1732
   .5850           -0.2079
   .5750           -0.2440
   .5250           -0.2512
   .4750           -0.3200
   .4250           -0.3607
   .3750           -0.4036
   .3250           -0.4490
   .2750           -0.4973
   .2250           -0.5498
   .1750           -0.6047
   .1250           -0.6663
   .0824           -0.7273
   .0498           -0.7759
   .0248           -0.7852
   .0113           -0.6703
   .0030           -0.2892
   .0000            0.7755
   .0030            0.8716
   .0113            0.5267
   .0248            0.2244
   .0498            0.0042
   .0824           -0.1113
   .1250           -0.1749
   .1750           -0.2051
   .2250           -0.2127
   .2750           -0.2090
   .3250           -0.1987
   .3750           -0.1845
   .4250           -0.1679
   .4750           -0.1499
   .5250           -0.1310
   .5750           -0.1116
   .6250           -0.0915
   .6750           -0.0707
   .7250           -0.0486
   .7750           -0.0246
   .8450            0.0177
   .9350            0.0997
   .9900            0.2438
```

flow is supercritical

```
IT=   0    RNORM=    ...
IT=   1    RNORM=    ...
IT=   2    RNORM=    ...
IT=   3    RNORM=    ...
IT=   4    RNORM=    ...
IT=   5    RNORM=    ...
IT=   6    RNORM=    ...
IT=   7    RNORM=    ...
IT=   8    RNORM=    ...
IT=   9    RNORM=    ...
IT=  10    RNORM=    ...
IT=  11    RNORM=    ...
IT=  12    RNORM=    ...
IT=  13    RNORM=    ...
IT=  14    RNORM=    ...
IT=  15    RNORM=    ...
IT=  16    RNORM=    ...
IT=  17    RNORM=    ...
IT=  18    RNORM=    ...
IT=  19    RNORM=    ...
IT=  20    RNORM=    ...
IT=  21    RNORM=  0.2408046E+01
IT=  22    RNORM=  0.2246656E+01
IT=  23    RNORM=  0.2330694E+01
IT=  24    RNORM=  0.1941462E+01
IT=  25    RNORM=  0.1363513E+01
IT=  26    RNORM=  0.1821728E+01
IT=  27    RNORM=  0.2700492E+01
IT=  28    RNORM=  0.2116095E+01
IT=  29    RNORM=  0.6764420E+00
IT=  30    RNORM=  0.1273140E+01
IT=  31    RNORM=  0.1474591E+01
IT=  32    RNORM=  0.1468042E+01
IT=  33    RNORM=  0.1640067E+01
IT=  34    RNORM=  0.2008251E+01
IT=  35    RNORM=  0.1940811E+01
IT=  36    RNORM=  0.1484638E+01
IT=  37    RNORM=  0.1025622E+01
IT=  38    RNORM=  0.8153395E+00
IT=  39    RNORM=  0.1601210E+01
IT=  40    RNORM=  0.1930673E+01
IT=  41    RNORM=  0.1426595E+01
IT=  42    RNORM=  0.6068323E+00
IT=  43    RNORM=  0.7080514E+00
IT=  44    RNORM=  0.9372849E+00
IT=  45    RNORM=  0.1206310E+01
IT=  46    RNORM=  0.9932569E+00
IT=  47    RNORM=  0.7025645E+00
IT=  48    RNORM=  0.5083519E+00
IT=  49    RNORM=  0.3186685E+00
IT=  50    RNORM=  0.1975036E+00
IT=  51    RNORM=  0.1107077E+00
IT=  52    RNORM=  0.1336054E+00
IT=  53    RNORM=  0.1131977E+00
IT=  54    RNORM=  0.8823371E-01
IT=  55    RNORM=  0.7365143E-01
IT=  56    RNORM=  0.6509942E-01
IT=  57    RNORM=  0.6424731E-01
IT=  58    RNORM=  0.6961179E-01
IT=  59    RNORM=  0.5650729E-01
IT=  60    RNORM=  0.3305346E-01
IT=  61    RNORM=  0.2351308E-01
IT=  62    RNORM=  0.9466037E-02
IT=  63    RNORM=  0.2124250E-01
IT=  64    RNORM=  0.2522635E-01
IT=  65    RNORM=  0.2406329E-01
IT=  66    RNORM=  0.2146107E-01
IT=  67    RNORM=  0.2598482E-01
IT=  68    RNORM=  0.2639437E-01
IT=  69    RNORM=  0.2135819E-01
IT=  70    RNORM=  0.1313454E-01
IT=  71    RNORM=  0.9345889E-02
IT=  72    RNORM=  0.5820463E-02
IT=  73    RNORM=  0.4675984E-02
Converged results IT= 73      RNORM=    0.0047
```

SUPERSTABLE FLOW RESULTS

lift coef CLE 0.55218
drag coef CDE -0.00031
moment coef about X= 0.10 is CM= -0.15236

YSI(T)	YP(T)	Q(T,1)	RMACH(T,1)
1.8232	0.0754	0.9620	0.7185
1.1119	0.1685	0.9140	0.6792
0.9900	0.3929	0.7867	0.5779
0.9350	0.4020	0.7976	0.5861
0.8450	0.1801	0.9079	0.6754
0.7750	0.0416	0.9772	0.7310
0.7250	0.0031	0.9945	0.7187
0.6750	0.0042	0.9955	0.7462
0.5750	0.0606	0.9079	0.7482
0.5250	-0.5505	0.9695	0.7247
0.5250	-1.1759	1.2647	0.9822
0.4750	-1.2239	1.5638	1.2817
0.1250	-1.2712	1.5978	1.3080
0.1750	-1.2890	1.6122	1.3352
0.3250	-1.3000	1.6213	1.3455
0.2750	-1.3003	1.6269	1.3518
0.2250	-1.2893	1.6271	1.3520
0.1750	-1.2669	1.6215	1.3457
0.1250	-1.2088	1.6101	1.3328
0.0921	-1.1351	1.5807	1.3002
0.0498	-0.9894	1.5489	1.2655
0.0248	-0.8835	1.4728	1.1853
0.0113	-0.6407	1.4220	1.1337
0.0030	0.1389	1.3071	1.0219
0.0030	1.0493	0.9293	0.6917
0.0113	0.9616	0.2772	0.1979
0.0248	0.6084	0.3820	0.2737
0.0498	0.2809	0.6614	0.4811
0.0824	0.0146	0.8541	0.6311
0.1250	-0.1346	0.9927	0.7439
0.1750	-0.2390	1.0664	0.8061
0.2250	-0.2952	1.1169	0.8495
0.2750	-0.3158	1.1438	0.8732
0.3250	-0.3162	1.1537	0.8818
0.3750	-0.3001	1.1538	0.8820
0.4250	-0.2758	1.1462	0.8752
0.4750	-0.2455	1.1346	0.8650
0.5250	-0.2155	1.1201	0.8523
0.5750	-0.1797	1.1056	0.8398
0.6250	-0.1509	1.0883	0.8249
0.6750	-0.1215	1.0743	0.8128
0.7250	-0.0809	1.0600	0.8006
0.7750	-0.0560	1.0401	0.7837
0.8450	-0.0176	1.0278	0.7733
0.9350	0.0416	1.0088	0.7573
0.9900	0.1801	0.9791	0.7326
1.0100	0.3929	0.9079	0.6743
1.1119	0.4020	0.7976	0.5863
1.8232	0.1685	0.7867	0.5778
	0.0754	0.9140	0.6792
		0.9620	0.7185

total execution time ITOTAL= 140897millisecs

ADDENDUM

The Relation Between the Number of Mean Line Elements, Internal Singularities and Grid Points

If the mean line is divided in n segments the following relation hold:

Number of distributed internal sources   =   n

Number of point sources (fixed)   =   2

Number of vorticity unknowns   =   n + 1

Number of point vortices (fixed)   =   1

Assuming that the number of grid points chosen beyond the trailing edge is n_w, the number of grid lines in the i-direction is $2(n + n_w) + 3$. If m levels are used in the j-direction we have for the lifting problem,

Total number of field panels   =   Total number of grid point

$$= m \times \{ 2(n + n_w) + 3 \}$$

For all airfoils except the NACA 64A410 airfoil, $n_w = 3$. For NACA 64A410 airfoil, $n_w = 0$.

For supercritical lifting flows past NACA 0012 airfoil, n = 21 and for NACA 64A410 airfoil n = 22.

For symmetric flows, there is no vorticity distribution on the mean line. Hence,

Number of distributed internal sources   =   n

Number of point sources (fixed)   =   2

Number of grid lines in i-direction   = $n + n_w + 2$

Number of grid lines in j-direction   =   m

Total number of field panels   = $m \times (n + n_w +$

For super-critical flow past the NACA 0012 airfoil, with

26 x 6 grid and n_w = 3, we have n = 21.

For sub-critical flows in general, the number of interna

singularities required are lesser since in this case close spac

of the i = constant grid lines is not necessary.